SCHOOL AND HOME GUIDE TO THE APPLE MACINTOSH COMPUTER

American Systems Computer Institute, Inc.

Everett Murdock / Susan Sudbury

Prentice-Hall, Inc.  Englewood Cliffs, New Jersey 07632
Libary of Congress Cataloging in Publication Data

MURDOCK, EVERETT (date)

School and home guide to the Macintosh computer.

Includes index.

I. Macintosh (Computer) I. Sudbury, Susan.
II. Title.
QA76.8.M3M87 1985 001.64 85-530

Editorial/production supervision: Karen Skrable Fortgang
Interior design: Linda Benson
Cover design: Photo Plus Art
Manufacturing buyer: Barbara Kelly Kittle

Macintosh is a trademark licensed to Apple Computer, Inc.
Apple is a registered trademark of Apple Computer, Inc.

© 1985 by Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publishers.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-793605-2 01
CONTENTS

PREFACE  vii

1  INTRODUCTION: THE MACINTOSH
A NEW STANDARD?  1
The Macintosh Difference  2
Innovations in Software  3
MacUsers  4
Macintosh Versus the IBM Standard  4
A Macintosh Standard?  7
The Macintosh and Educational Computing  8

2  BUYING A SMALL COMPUTER  12
The "Which One is Best" Question  13
Hardware Considerations  16
The Speed Question  19
The Memory Question  21
The Money Question  24
Summary  26
# Contents

## 3

**A CLOSER LOOK:**

**THE HARDWARE** 27  
*Parts and Pieces* 27  
*The Keyboard Versus the Mouse* 28  
*The Mouse* 29  
*Mouse Functions* 31  
*The Keyboard* 32  
*Typewriter Keys* 32  
*Special Keys* 33  
*Enter Key* 33  
*Option Key* 33  
*Command Key* 34  
*Software Usages* 35  
*Numeric Keypad* 35  
*Disk Drive and Diskettes* 36  
*Back Panel* 38  
*Inside the Box* 38  
*The System Board* 38  
*The Analog Board* 39

## 4

**A CLOSER LOOK:**

**THE OPERATING SYSTEM** 40  
*An Easy-to-Use System* 41  
*The User Interface* 42  
*The Nature of Operating Systems* 43  
*Early Operating Systems* 44  
*The New Generation OS* 46  
*Apple's New OS* 47  
*The Desktop Concept* 49  
*Functions* 52  
*The Desktop* 53  
*Programs and Windows* 53  
*Trash Feature* 57  
*Built-In Programs* 58  
*The Future* 63

## 5

**A CLOSER LOOK:**

**MICROSOFT BASIC** 64  
*Macintosh BASIC* 64  
*Using "Incompatible" BASIC Programs* 65  
*How Useful is Basic?* 66  
*Learning a Programming Language* 67
6

A CLOSER LOOK:
COMMUNICATIONS AND
MODEMS 74
Buying a Modem 75
Modem Standards 75
Speed 77
Communications Software 79
Software Features 80
Some Modem History 81
Reach Out 83
How Do I Use a Modem? 83
About Bulletin Boards 85
Consumer Databases 86
Educational Databases 88

7

EDUCATION: THE MACINTOSH
AT HOME 90
Finding the Best Place to Keep the Macintosh 91
The First Encounter 91
Who's Teaching Whom? 92
Computers and Confidence 92
Working Out a Timesharing Plan 92
Creating a Home Curriculum 93
Home Education 94
Personal Computers and Your Preschooler 95
You Don't Have to Read to Use It 96
A Bridge Between a Book and a Toy 96
Abstract to Concrete 96
Educational Software for Preschoolers 97
Start an Educational User's Group 97
Become an Advocate for Computer Literacy in Your Community 98
Educational Considerations for Older Children 98
Encouraging Math Skills in Girls 99
Teach Them to Type 99
Programming: A Valuable Skill 99
A Word Processing Program as an Educational Tool 101
Training For College Entrance Exams 102
Modems: A Link to the World 102
Summer Computer Camps for Kids 103
Contents

8
EDUCATION: THE MACINTOSH AT SCHOOL 105
Questions to Ask Yourself 106
Why Do We Need Computers in the School Classroom? 106
What Do We Want the Computer to Do? 110
What Hardware and Software Will We Need to Meet Our Teaching Objectives? 126
What Will It Cost to Purchase Computer Equipment? 131
Where Can We Find The Money to Buy Computers for Our School? 133
Where Will We Put The Computers? 136
Who Will Teach The Teachers? 136
How Do We Go About Selecting the Best Educational Software? 137

9
MACINTOSH SOFTWARE 144
Software Development Strategy 145
The Future 145
New Software for the Mac 146
More Future Software 151
Software Under Development 152

A
EDUCATIONAL COMPUTER MAGAZINES AND JOURNALS 168

B
EDUCATIONAL COMPUTING 175

C
ASSOCIATIONS FOR EDUCATION AND COMPUTING 182

D
SOFTWARE REVIEW CENTERS 185

E
COMPUTER CAMPS 187

F
COMPUTERIZED CLASSROOM PLANNING SHEET 190
GLOSSARY 195
INDEX 201
The Macintosh, Apple Computer's most innovative personal computer, has been called the computer of the future, the computer we have all been waiting for. Apple advertises it as the computer "for the rest of us," meaning that all the other personal computers that have been on the market up to now were designed for "computer people." Every Macintosh advertisement makes the same point: Until now, computers have been hard to use, but using the Macintosh is easy.

But if the Macintosh is really the computer of the future, why are some knowledgeable people saying it is a flawed machine? Some even predict that in the IBM-driven world of personal computing, the Apple Macintosh has a less-than-even chance of surviving. Is the Macintosh really the computer of the future, a landmark design that will dictate the direction for all future computers—or will the Macintosh even have a future? Will it end up on the scrapheap of history as a clever little box that for a while was made by a company called Apple that nobody remembers anymore?

In this book we examine such questions, using all of the available evidence to help us find the answers. In the process, we try to be fair and impartial. We have studied computers and educational computing for many years, and we have written books about other computers describing their strong and weak points; therefore, we look at the Macintosh specifically as to how it compares to other personal computers. We have not approached our study of the Macintosh as proponents or detractors. We believe that any computer product has a potential role in the educational computing environment: Maybe the Macin-
tosh is not as well suited to the classroom as one of the IBM personal computers. Or perhaps Apple’s reliable classroom mainstay, the Apple II family of computers, is still the best choice. On the other hand, some see the Macintosh as the salvation of both home and school computing. In light of the evidence, we see it as falling somewhere in between. In this book we will present our reasons.

What about the Macintosh as a home computer? Will Apple encourage the people who write computer programs for home use to write them for the Macintosh? The Apple II series of computers got a lot of use as preschool trainers by parents who wanted to introduce their children to computers early on. Will those learning games be modified to run on the Macintosh?

In this book we look closely at the first of the learning programs that have been released for the Apple Macintosh. We evaluate their merit and their usefulness in school and at home. We also have made an effort to learn what computer programs are in the works. Apple Computer Inc. has been very helpful to us in this regard. And we have explored other sources to determine what programs will eventually be available for use with the Macintosh.

We also try to give the reader a clear idea of just what is so different about the Macintosh. Before the Macintosh was generally available for purchase, Apple provided us with a computer and a variety of programs. We spent endless hours playing with the Macintosh, trying to learn its secrets, its idiosyncracies, its not-so-obvious strengths and weaknesses. We have made every effort to report these fairly and honestly.

This book describes the operating system used on the Macintosh as well as an overview of the hardware design. We compare the Mac’s design with that of other popular personal computers, and we examine how these design choices affect the use of the computer. We present a brief description of BASIC programming commands and the particular implementation of BASIC used on the Macintosh, and also look at the role of telecommunications and modems.

WHO SHOULD READ THIS BOOK?

In the following pages, we look closely at the design and usefulness of the Apple Macintosh personal computer; we also examine the implications of its design philosophy. We discuss the impact—realized and potential—of this new computer design in varied educational settings.

This book is for anyone who wants to know more about the Apple Macintosh. But it is especially for anyone who wants to know more about using the Macintosh in education. It is for anyone who wants to know how to use the Macintosh and its computer programs as educational tools.

Our evaluation includes—but is not restricted to—the uses of this computer in a school setting. The Macintosh is also making its way into the home. More and more home-oriented programs are appearing every day. Learning games and home education programs are among the top-selling computer programs worldwide, and this is a market that Apple will not overlook.
In this book, we have included a list of every educational program for the Macintosh that we could find. We have evaluated as many of these home/school-learning software packages as we could get our hands on.

Developing learning software is one of the fastest growing business enterprises in the United States. Not all of the new software is good, but much of it represents the cutting edge of innovation in education. We examine each program and ask: Does it do what it promises to do? Does it utilize the computer's attributes? Does it break new ground in the search for effective teaching methods?

And if the chapters on educational software can be viewed as taking a hard look at what kind of job these programs do, the rest of the book should be seen as taking the same kind of hard look at the effectiveness of the hardware that runs them, the Macintosh itself. Is this machine useful as a learning tool? Is it cost effective? How can it best be used in school? At home? We try to answer these questions and many more. So come along with us as we explore the Apple Macintosh. We look at the main components of this computer, both in overview and in detail. But we don't stop there. We are educators and we are interested in where education is going in the future. We therefore sometimes lean to speculation. We weigh the historical trends of a rapidly changing computer world and try to surmise where it is leading us. Apple has established itself as the world leader in educational computing. The future of Apple and of educational computing itself is closely tied up with how well accepted this new computer will be. Computers are rapidly becoming a part of all of our lives. We believe the Apple Macintosh has an important role to play in determining where educational computing is heading. Join us as we examine what that role may be.

EASE OF USE

It was our intention to make this book easy to read. It was designed to be easy to work your way through on the first reading, and then easy to find a specific topic if you want to go back later and reread something.

To make it easier to find topics, we have printed the name of the subject being discussed at the top of the right-hand page. The chapter title appears at the top of each left-hand page.

Two other ease-of-use features appear at the end of the book. An index to the book's contents has been constructed to respond to the text-searching behaviors we have observed in both beginners and experts. That is, beginners will search for a topic in terms of function, so we have included functions topic words; but an expert, someone more familiar with the technology of computers and computer terminology, will search for names and titles, so the index also includes all the names of products, features, and terms related to the Macintosh and educational computing.

There is also a glossary which includes brief definitions of words or phrases that may not be familiar to some readers. Defined words will be italicized soon after they are introduced. This is to remind you that the definition of these words can be found in the Glossary. If the definition is complicated, or if it is key to the points that follow, it is set aside and marked off from the rest of the text.
For example, we could pause here to define the word *computer*.

---

**Definition: Computer**

An electronic device which receives input (of various types, but most commonly your input from the keyboard), puts that input into storage, operates on that input as directed by a computer program (often called software), and outputs the results. This output could be to command a robot in a factory, to guide a rocket, or to spell out a message on a display screen; whatever the output, its purpose and utility is always directed by you, the user.

---

We also occasionally include other special sections, set off from the rest of the text. These may be technical notes or special tryouts. These sections are designed to briefly describe a technical feature or some other unusually interesting aspect of the Macintosh. These notes will be of special interest to those who are seeking a more technical knowledge of the computing topic under discussion.

**CHAPTER SUMMARY**

1. **Introduction: The Macintosh. A New Standard?**. An overview of the Macintosh and an analysis of how it fits into the changing world of computers. This chapter looks at emerging directions in educational computing.

2. **Buying a Small Computer**. This chapter is based on a series of questions the buyer of a personal computer might (and should) ask.

3. **The Hardware: What Makes a Macintosh?** This chapter looks inside the box to see what design decisions were made to develop this innovative computer. It answers the question, What is different about this computer from all the others?

4. **The Finder: A New Type of Operating System**. This chapter provides an in-depth look at the soul of the Macintosh, its operating system.

5. **Using BASIC**. Here we present an overview of the Microsoft BASIC programming language offered on the Macintosh. This chapter also describes some of the common commands and how you might use them.

6. **A Closer Look: Communications and Modems**. This chapter looks at the details of the growing world of computer communications.
7. EDUCATION: THE MACINTOSH AT HOME. In this chapter we discuss ways that you might use the Macintosh for your children's education; we suggest activities for kids of all ages, whether they be preschoolers first learning about the world of computing or older children looking for expanded uses of computers or preparing for their college entrance exams.

8. EDUCATION: THE MACINTOSH AT SCHOOL. When computers are brought into the classroom, new and complex issues must be considered. In this chapter we will discuss such issues; among the topics we examine are finding funding for computers, informational and computer-aided learning resources, and how to select software.

9. EDUCATIONAL SOFTWARE REVIEW. In this final chapter we provide a list of educational software presently available for the Macintosh.
INTRODUCTION:  
THE MACINTOSH—
A NEW STANDARD?

Every time you turn the Macintosh on, the words

WELCOME TO MACINTOSH

appear on the computer’s screen. What may not be apparent to those who have never used any other computer except the Macintosh is that these three words represent a revolution in personal computing.

When you turn on most computers designed before the Macintosh, you get a screen message that looks something like this:

ACME COMPUTER COMPANY DOS
VERSION 1.11
COPYRIGHT 1984, 1985

CURRENT DATE: 12-3-84
ENTER CORRECT DATE:

And if you type in the correct date and time, you get a screen that looks like this:

A>
This is the dreaded A> prompt. There has probably never been a computer screen message that has struck so much fear into the hearts of so many computer users as the A> prompt. The problem is, the computer doesn't help you. You are supposed to know what to do. You are supposed to know a disk operating command to type in next to the A>. You have to know exactly the right spelling and exactly the right order and exactly the right punctuation. Get it wrong and you will get a terse error message. After you get the correct message keyed in, you have to press the right key in order to make anything happen. And, of course, you had to know the correct way to insert the operating diskette into the drive slot or you wouldn't even have gotten the opening messages and the A> prompt in the first place—you would have seen only a mysterious number and a cryptic error message.

THE MACINTOSH DIFFERENCE

But look at the Macintosh! If you turn on the Macintosh without a diskette in the drive, the computer shows you this picture:

This picture of a diskette represents the big difference. Most computers are designed to be operated by people who know how to operate computers. But what if you don't know? The Macintosh is designed for those who don't know. Apple Computer Inc. has designed and built this new computer as a machine to be used by those who don't necessarily know how to use a computer.

But is this the whole story? Is it just a new computer easier to use than the computers that came before it? Not by a long shot. This computer will be purchased and used by a great many people who are very familiar with computers. The Macintosh not only presents an innovative design that will make it easier to begin using the machine, but they also created a machine that is faster and designed to utilize innovative—often startlingly innovative—computer programs that act as if they were built into the computer.

Computer programs (collectively called software) are what makes a computer capable of doing a task. Someone thinks up a useful task (say adding up numbers or organizing memos or drawing a picture on the computer's screen). Next he or she writes a computer program designed to do that task. That software writer (called a programmer) can also write into the program helpful things that make the computer program easy to use. And
that is what the people who wrote many of the Macintosh’s main programs did; they thought up a variety of new uses for their new computer and they made them all very easy to use. Doesn’t seem like such a big deal? It wouldn’t seem so, but a lot of people have been trying for a lot of years to do it. A computer is not a toaster. It is a highly complex tool that has gradually evolved into a more intricate and durable design. No matter what the ads say, a computer is a lot harder to use than a toaster or a TV set or an electric oven, because most computers, unlike appliances, are not preprogrammed. Your home stereo is designed to do one basic task; amplify sound. A computer may also be capable of amplifying sound. It can also supervise other noncomputer tasks such as cooking your supper or monitoring the performance of your car’s engine. But to do these tasks, it has to be programmed. A single-purpose computer is no harder to use than a washing machine; you just turn it on and it runs. Some washing machines even have little dedicated microprocessors in them that work like that. But with a personal computer, you have to tell it what to do, and that’s where it becomes complicated. Usually you have to read a lot of complicated documentation before you even begin using a computer. With the introduction of the Macintosh, Apple has tried to change all that.

INNOVATIONS IN SOFTWARE

Up to now, no one has been able to make a computer that could teach its users how to use it. Apple has not taken the final step down this self-teaching path, but they have taken a big step. It has provided us with a direction that will no doubt spur a lot of research into how best to make tasks easier to accomplish in future computers.

And Apple has done something else with this new computer that for some reason doesn’t get as much attention as all the user-friendly programs: it has provided a new class of software—innovative programs such as MacPaint and MacWrite—that give the user a variety of new computing options. These programs are tied into the basic design of the computer’s easy-to-use operating system, and they combine to turn the Mac into a valuable productivity machine that will undoubtedly find a place in every school or office. Using this new software, even an untrained user can create detailed charts, diagrams, and artistic creations that can be combined with text to make professional-looking printouts. If a school or business’s art and graphic design department doesn’t get a Macintosh, the people who have to make graphic presentations will be going out and getting their own.

These new programs are so easy to use, so remarkably advanced compared to most computer software, that experienced computer users may become more interested in how they were written than in what they can do. They are so well integrated with the design of the whole Macintosh system that it may seem to the new user that these programs are the Macintosh. People describe the Mac by describing what MacPaint does. They say the Macintosh is a machine that can fill in drawings with something called the Spray Can. When they are telling others why they like the machine that is the Macintosh, they describe MacWrite’s variety of ornate fonts; they often point to the program’s ability to combine text and graphics.
MACINTOSH VERSUS THE IBM STANDARD

In its ads for the Macintosh, Apple has openly challenged its main opponent in the small computer arena, IBM, with its ever-expanding family of Personal Computers. But Apple’s main opponent is not really the hugely popular IBM PC. To be successful in the personal computer market, Apple must contend with the IBM PC standard.

And what is the IBM Standard? It is a particular, small computer design that has sold so well, and attracted so many software producers, that it has spawned a large following of other computer makers who have designed and built similar computers.

When the IBM PC appeared in 1981, it probably seemed to many like just another one of those funny little computers. Back then, there were only a few manufacturers of these small machines and their sales were measured in the mere thousands. Radio Shack was seen as the computer hobbyist’s center, a place that sold computers as a side-line. Their customers were often thought of as “computer whizzes.” Other companies, like Apple and Texas Instruments, were fighting it out for the home computer market. Meanwhile, IBM was making the “real” computers: it made big computers (called mainframes) that ran America’s big businesses. No one expected giant IBM to even notice that there was such a thing as a personal computer, let alone develop one.
But, late in 1981, IBM made the surprising announcement that it was about to begin selling a low cost, personal computer with one disk drive and 64K of user memory.

The new PC was a big success from the first day it appeared in the stores. It created a whole new class of computer customers, businessmen, and professionals who had wanted all along to join the computer revolution, but who couldn't see how the typical home computer could help them. The availability of a relatively cheap little computer with a lot of power and IBM's trusted name signaled the start of small business's conversion to computer power.

Despite the impact the PC was having in the business world, most of the other small computer makers were still pursuing the old strategy of designing individually unique and esoteric computers. But growing IBM PC sales were beginning to cut into their profit margins. They took note of the PC's relatively straightforward hardware design, and they soon realized that the reason for the expanding PC sales lay in the involvement of third-party software makers. Third-party companies are in the business of providing what goes in between the two main parties in any transaction, the buyer and the seller. They try to determine what is selling and then look carefully at what they could provide to augment the hot sellers. All of these small software companies closely watch the new trends in the computer business; their business lives depend on it. When these third-party businesses see a new computer that is beginning to catch on, they swing into action quickly. And none of them had ever seen anything as hot as the IBM PC. So they all started making new products for it. They got behind the IBM PC to the extent that soon anything a user might want, from home applications to business to educational programs, was available for the IBM PC. And this, in turn, made the PC sell even faster.

At the same time the IBM PC was finding such success, there was general public dissatisfaction with what was going on in the small computer world. Each manufacturer was producing both the specialized hardware and the software packages designed to run on a particular machine. There was no way to carry software from one machine to the other. Everyone was calling for a standard but no agency or organization seemed to have the clout to create it. The IBM PC did not introduce a new or innovative design, but it did introduce a new concept: Whereas past computer manufacturers had held their system's design and operating system as a guarded secret and tried to control the software aftermarket, IBM made the PC architecture and operating system public. The key to the concept was that IBM encouraged software development both officially and unofficially.

The final step, as always, depended on sales. If IBM's PC became a big seller, it would be profitable for the third party producers to make software and add-on hardware for it. If not, then the small software companies could safely ignore the PC and stick with a conservative expansion philosophy that responded to the needs of the three leaders, Apple, Atari, and Radio Shack. But the PC did sell. It sold not only because of the IBM name, but also because it was of a good, solid design. And IBM discovered that advertising didn't hurt. It advertised the PC like no computer had ever been advertised before.

When some other small companies came out with machines that were virtual clones of the PC, everyone wondered what IBM would do about it. In the past IBM had been quick to take other companies to court if they came too close to proprietary IBM designs. And a protracted court battle with IBM was not something a small company would look forward to. But this time IBM sat back and watched. As a result, the PC clone manufac-
Manufacturers began to feel safer. Others jumped into this lucrative market (including Radio Shack and some of the other established companies) and by the middle 1980's there were more than fifty PCs on the market advertising themselves as IBM PC compatible.

For the first time, a variety of manufacturers were all promoting one standard, the IBM PC standard. Now they all had a stake in it.

Most of the larger computer companies gave in. The emerging standard was beginning to dominate the computer industry. As sales of all the non-IBM compatible computers slowed, even big companies like AT&T, Sperry, and Texas Instrument were forced to come out with IBM PC clones. Even Radio Shack eventually jumped in with its IBM PC compatible Model 2000. And what was Apple doing? A few years earlier, it had introduced a new line of powerful (and expensive) computers, the Lisa series. These computers were based on the very powerful Motorola 68000 central processing system and offered their users an entirely new type of access to their programs. Based on a system of pictorial messages (called icons) and a simulated desktop working environment, these computers were very easy to use but were too expensive for the average user.

This is where Apple ran head-on into the IBM standard. The Lisas were introduced just when the IBM standard was gathering momentum. At $10,000 apiece, the Lisa computers could only be a justifiable purchase for businesses. But businesses were going to the IBM standard in droves. Both the IBM PC and its compatibles were being sold as workstations to be attached to company mainframes. The Lisa just didn't fit in. Despite generally positive reviews, the Lisa wasn't selling. And as a result, Apple was in trouble. Its main source of income was still the useful but aging Apple II series of computers. Computer business watchers waited to see how Apple would solve its problem. Would it try to upgrade the Apple II series? Would it try to sell a lower-priced Lisa? Or would it—like most of the others—bow to the marketing pressure being generated by the IBM standard? There were some that said that the latter choice offered the only hope for the company's survival.

But the decision had already been made. Apple was secretly developing an entirely new computer, a computer that took the best of the Lisa's user-interface technology and combined it with a bundle of highly innovative operating programs. It was packaged all together within a small case of an entirely new design. A carefully redesigned electronics system and a brand new high-speed manufacturing facility allowed Apple to lower the price of the Lisa methods to an affordable range. Thus the Macintosh was born.

The next step depended on sales. If the new computer didn't catch on right away, if it couldn't directly challenge the predominance of the IBM PC standard and win, the very survival of the company was in doubt. But it did catch on. Despite the new manufacturing plant, Apple couldn't keep up with the demand for Macs. The company announced that 70,000 Macintoshes had been built and sold within the first few months of its existence. They were backordered everywhere.

One reason for the early brisk sales, was a special deal Apple made with a number of universities across the country. Apple promised to deliver Macintoshes to these schools in quantity, at a drastically reduced price. All the schools had to do was agree to use them in computer classes and encourage the development of new Macintosh software programs for education. This was a masterly stroke of marketing strategy because it not only pro-
vided a good base for early sales, but it put the Macintosh—a new technology machine—into the hands of computer students who could appreciate that technology and pass the word. The Mac became well-known to students, even if the limited supplies kept it from being widely accepted in the outside world. At the University of Utah, where we teach, it seems as if everyone has one. You see the distinctive Macintosh 3½-inch diskettes lying around everywhere.

A MACINTOSH STANDARD?

But to dislodge the IBM standard, to create a new standard that will compete for sales with the IBM PC and its compatibles, the Macintosh will have to do more than sell a lot of computers; the Macintosh will have to catch the fancy of the third party software developers. A standard implies widely available programs. In order to attract a large base of programs that will add to the utility of the Macintosh, Apple will have to both encourage and allow software development. It will have to be willing to help software developers modify their programs to run on the Mac. And it will have to allow these third party software producers the right to stake out their own territory. By that, we mean that Apple will have to keep its hands off certain types of programs. If Apple internally develops a large variety of applicable programs, it will be adding to its pocketbook at the expense of the third party developers. Specialized application programs are difficult and expensive to develop. No software maker will be willing to spend the time and money to create a program if Apple is going to unexpectedly announce its own version. So far Apple has kept its hands off. And it has set up a working relationship with a group of designated Apple software suppliers.

The Macintosh presents program development problems that other makers don’t have to contend with: The unique operating system of the computer requires a unique interaction with applications programs using Apple’s set of pictorial on-screen messages and file management methods. The “mouse” pointing device also presents some difficulties to software makers accustomed to the cursor key system. Possibly due to these limitations, as well as the relative complexity of the Mac’s hardware design, new applications programs were at first slow to emerge. But within six months of the computer’s introduction, new programs were being rapidly developed. And the longer the Mac has been on the market, the more we are seeing new approaches to computer tasks. The IBM-standard operating environment has dictated a certain narrow approach to programming design; now the new operating system of the Macintosh is spurring new options in programming design. It is as if the complexity of Macintosh’s user interface has convinced programmers that their programs needed more than a face lift—their programs need to be entirely redesigned.

So, will the Macintosh create a new Mac standard to challenge the IBM PC standard? It may be some time before we can be sure. Surely, part of the answer is in sales, and the Macintosh is selling well. Another key is the support the computer is getting from the third party software developers. This also seems to be going Apple’s way. Finally, the development of an alternative standard is dependant to some degree on what
IBM does in response to the challenge. IBM's newest personal computer, the PC-AT, has taken a large step forward by using a faster and more powerful processor while still allowing a lot of the existing software to run. This will make it tougher for Apple to establish a foothold.

There is a rule of consumerism that applies to computer technology just as it applies to any other retail product: If the buyer wants it and is willing to pay for it, it will be produced. The more the mouse pointing devices and pull down, on-screen menus used on the Apple Macintosh catch on with the buying public, the more the PCs and the PC-compatibles (and all other computers for that matter) will look like the Macintosh. They too will begin to use mouse pointing devices, integrated (several programs combined into one) software, and other programming techniques that make the computers more accessible to the user. However, just because computers begin to look more like the Macintosh does not mean that there will be a Macintosh standard. Such a standard (and the resulting boom in Macintosh-compatible computers) would have to be the result of something similar to the complex mix of factors that led to the formulation of the IBM PC standard. One very important factor was that IBM both allowed and encouraged the compatibles to emerge. Whether Apple or any other manufacturer will do the same is only conjecture.

THE MACINTOSH AND EDUCATIONAL COMPUTING

A great deal of Apple's success with the Apple II series of computers was due to the acceptance of the computers in the public school systems. A question of concern to educators, therefore, is just what Apple intends to do with the Macintosh. Will the Apple make as big an effort to establish the Mac in the public schools as it did with the Apple II's? Will it encourage the modification of the huge base of educational software for the II series to make these programs run on the Mac? We suspect that Apple will. The university consortium arrangement that Apple set up when introducing the Mac is an indication of Apple's commitment to the educational market. But the Apple II software presents a problem. All software made for the Apple II's (including the IIc portable) is based on the old MOS 6502 processing system. This system is not at all compatible with the new Motorola 68000 system used in the Macintosh. Converting the software would involve a major rewriting of each program's code. A major problem with this rewriting effort is that a great many of these programs were written by small software concerns, often just individuals writing out of their own homes. It is an open question whether these programmers have the resources to rewrite their programs to run on the Macintosh. The introduction of the IIc portable indicates that Apple does not intend to abandon its still-strong sales base of Apple II products. This might be another factor that may encourage education software producers to stick with the Apple II programs. To a great degree, the future position of Apple computers in the educational environment depends on Apple. The company has a strong position in educational computing that could be maintained by a vigorous effort to create a large base of educational programs for the Macintosh. We will have to wait and see.

Since limited funds and limited teacher and parent training is a fact of life in the
**Technical Note: Central Processing Unit (CPU)**

The central processing unit (abbreviated as CPU) is the part of a computer that does the actual processing of data. In large computers, it may refer to a centrally located cabinet that contains the main processing circuitry, but in small computers the term usually refers to a single silicon chip somewhere inside the computer. Built right into these CPU chips are all the main operating circuits for handling the flow of data in and out of the computer. Each manufacturer of CPUs designs its chip in a slightly different way. This means that software producers must create programs that will "talk to" each type of CPU.

There are several types of CPU chips currently on the market, from various manufacturers. The 8088 CPU used in the IBM PC is made by the Intel Corporation and is also available for use in any other computer that wants to emulate the operation of the IBM. Intel also makes various other CPUs that have somewhat different methods of operation, but are to some degree compatible with the 8088. Motorola Corporation makes the central processor used in the Apple Macintosh. It is called the 68000.

CPU chips are often referred to as 8-bit or 16-bit or 32-bit. The bit refers to the way the CPU handles data. A bit is a simple on/off signal used by the computer to encode information. A string of these bits is used to make up a unit of information. For example, using a standard code (the ASCII code), it takes 8 bits to represent, in code, one letter or number typed in from the computer's keyboard. Early computers, therefore, chose 8 bits as the logical size of bit groups with which to handle incoming and outgoing data. But more modern computers, like the IBM PC and its compatibles, handle data in 16-bit chunks. This speeds up the process because the data is shunted around inside the computer 16 bits at a time instead of the usual 8. These days, computers are also referred to by their input/output bit-handling capability. The IBM PC, for example, takes care of its internal tasks by handling data in the 16-bit chunks we mentioned above. But when it comes time to take in information from the outside (from the keyboard, for example) or to send out some data (say to a printer), the IBM handles it in the smaller 8-bit chunks. For this reason the IBM PC is often referred to as an 8/16 bit computer. Some of the newer CPUs from Intel can handle data in all cases in the larger 16-bit chunks. These processors (and the computers that use them) are therefore referred to as 16/16 bit.

The Macintosh takes the next step in processing power. It moves on to the 32-bit handling standard. The Mac uses 16/32 bit processing based on the 68000 processors. This means it uses the 32-bit system to handle data internally, but is slowed to 16-bit methods for its external tasks. A few of the newer computers are using true 32/32 bit processing. This represents a data-handling capacity formerly reserved for the big mainframe computers that cost hundreds of thousands of dollars. Whether or not Apple can successfully lead the way to a new 32-bit industry standard is still open to debate, but the 32-bit CPUs are already available, and their cost is decreasing every year.
computer education field, the answers to the two questions lie somewhere in the future; school computing and home educational computing has not kept pace with business computing. Sad as that fact is, there seems to be little hope of its catching up, at least in the near future. This fact forces computer makers like Apple to consider carefully how much money and effort to put into what continues to be a market of secondary importance. Business consumes by far the majority of all the computers made today. Some businesses buy personal computers by the thousands these days, usually IBM PC or its compatibles. Apple is surely looking at ways to enter that market. The question is, will the Macintosh be advertised and sold as a business machine or a learning machine? We suspect that it will be sold as both, but, again, it will probably be the software producers that eventually determine the answer. If they get behind the Mac like they got behind the Apple II’s, we will eventually see the Macintosh in every classroom. If not, the IBM PC standard will probably dominate school computing just as it has taken over the business world.

Still, there is probably no area of software development that is growing as fast as the educational area. Home educational computing is growing as fast as any, in the form of computer learning games. Thus far, Apple’s ads seem to be keeping to a very general tack, not representing the Macintosh’s use in any one type of computing situation. IBM is pushing the educational market much more now than in the past, partly because of the introduction of its own low-cost IBM PC-compatible computer, the PCjr. Other compatibles, searching for new market outlets, are making high volume discount deals with schools and universities. It would not be surprising to see one of the IBM compatible manufacturers make school computing a priority market. It is a growing, viable market that will consume millions of dollars worth of computing hardware in the next few years as more and more schools see the necessity of training their students to deal with the technology that is already part of all of our lives. Apple, Atari, and Radio Shack, all more often thought of as makers of home computers, have spent a great deal of time, money, and energy in the computer learning market.

The Apple Corporation is a very small company compared to IBM, and a small corporation, with more closely held stock, can be more aggressive. It can test the forward movement of the market into new technologies. Apple is willing to do this, as indicated by its introduction of new machines the average user is not sure he or she is ready for: the ahead-of-its-time Lisa, and the Macintosh, which Apple sees as its leap-forward machine. The problem is that new-technology machines, by definition, lose their base of existing software with every jump forward. By sticking with the technology represented by the PC and the PCjr, IBM obviously believes the market will choose old software over new technology. How Apple decides to respond to IBM’s inroads into the educational market will affect not only the viability of Apple’s educational computing position, but also the technology and software development policies of other computer manufacturers.

A lot of people who like to speculate about the future of educational computing (including us) see IBM poised to make a major entry into the educational computing field. IBM has already set up model schools and classrooms around the country, supported by the donation of millions of dollars in cash and in computing equipment. But so far, IBM has not chosen to directly challenge the other manufacturers—especially Apple—in the schools. The introduction of the PCjr has made some inroads into the schools, especially
as workstations in classroom networks of small computers, but to date most PCjrs are still being sold into the same business base as the PC. The majority of educational software created in the past several years was designed to run on the old Apple II family of computers. These time-tested small computers have been the workhorses of the young field of educational computing. Apple has carefully built up its lead in the education field and it is about to give up that lead without a struggle.

One computer maker, Atari, is actively pursuing the educational computing dollar. It is still encouraging the development of educational software through its monetary reward program for new packages that will run on its lowest-cost home computers. And many schools have at least a few Ataris in place. It should be remembered that Atari is owned by the entertainment giant, Warner Communications, and that it is a continuing success story in the computer games market. Atari could use computer games as a base to widen its educational marketing strategy. The educational games market is just getting off the ground and could be a big winner for Atari in the future: A lot of parents will buy an educational game for their children where they wouldn’t pay a nickel for one of the "just for fun" games.

Radio Shack is also a presence in educational computing, especially in the home learning environment. And it is making a strong push into school computing as evidenced by its standing offer (frequently publicized in educational computing journals) to allow any bona fide teacher free admission to any of its computer courses. The idea, of course, is to familiarize teachers with the advantages of Radio Shack’s computers and software.

Questions that relate to the introduction of the Macintosh into the educational community—and the impact on education of new computer technology in general—will be discussed at much greater length in the chapters on education later in this book. There we look more closely at some of the specific educational computing issues that now confront parents and educators.
Buying a small computer is about as tough as buying a new car, except you don't even get to look under the hood. Not only that, but it wouldn't do you much good if you could look under the hood of a computer. All you would see is a maze of wires and circuit boards. And what those electronic parts do and the different methods every computer manufacturer uses to get the job done are so complicated you almost have to have a college degree in computers to understand it. Because of this complexity, and because of the constantly evolving electronic technology, many computer ads try to summarize the advantages of their product; they tell you how much memory it has, or that it runs fast. If this was all you could find out about a computer before you bought it, it would be like buying transportation for your family by only finding out how big the engine was, or what its top speed was out on the highway. You might want a car and end up with a truck, or a Formula 1 racing car. Before you buy a car you have to find out some of the basics, like what kind of fuel it uses and whether you can get that particular kind of fuel in your neighborhood; and you probably wouldn't care how fast it will go out on the race track if you only plan to use it to go down to the grocery store and back.

Since you can't learn everything about how every computer is built, the trick is to know the right questions to ask. So we have organized this chapter around the kind of questions people might be expected to ask if they were in the market for a small computer. And then we try to present the answer in a conversational way, also going over some of the things you might have not known to ask.

Let's start with a very general question.
THE "WHICH ONE IS BEST" QUESTION

All the ads claim that their computer is the best. How should I decide for myself which one is best?

Answer

If you ask any computer expert which computer to buy, he or she will almost always say,

DON'T BUY THE COMPUTER; BUY THE SOFTWARE.

What this well-worn phrase means is that the important decision is not which piece of "iron," (the hardware) to buy, but what you want to do with it.

The important thing is to ask yourself some questions before you go looking. What do you plan to use your computer for when you get it home? Each task you ask a computer to do requires a computer program to do it; in other words, you need software. Unless somebody also sells some software to go with your new computer, it might well become a very expensive paper weight. This is an especially important issue if you are looking at the Macintosh, because the Mac is one of the newcomers on the market. New computers present a problem: No one has had time to prepare any software for them. With a relatively new computer like the Macintosh you have to ask a different question: how much software will be available for it? The answer to this question takes some investigation. The first bit of information you should look at is how fast the computer is selling. If it is a hot seller—like the Macintosh—you can rest assured there will be plenty of software forthcoming. Next, you should investigate just what markets the computers are going into. This too is an indication of what software will be available. For example, if the Macintosh were being sold as primarily a computer-aided design (CAD) machine (it would do an excellent job of this type of computing), you could not expect any of the software developers to be creating educational software for it. Looking at it that way, the jury may still be out on exactly what type of software to expect for the Macintosh.

Clearly the Macintosh is an excellent "productivity" machine. That's what they call it around Mac headquarters in Cupertino, California. It is one of the best computers on the market for producing hard copy, as printed computer output is called. With the Mac, you can create very professional-looking manuscripts and can include artwork, charts and tables, and various complex printed designs with printed text. You can even produce professional-quality engineering designs and schematics in high enough quality reproductions to be turned into overhead projection slides by a photocopy machine. These kinds of applications are within the capability of the Macintosh's main applications programs, MacWrite and MacPaint. So, what kind of computing environments is this computer likely to end up in? Well, who couldn't use that kind of output? Businesses will certainly find it useful. Planners of all types—from architects to contractors to engineering designers—will find many uses for it. And what about teachers who have to design multimedia lesson plans? And business planners or account executives who need to make professional-looking and saleable presentations? It seems likely that just using the basic software that Apple provides will guarantee a wide variety of users. This is the first step in
getting a new computer accepted by those who create software. The evidence is that each of these factors is leading to a large software base for the Macintosh.

To try to get a better idea of what is in the software future for the Macintosh, we visited Macintosh development headquarters in Cupertino, California. Apple gave us a large list of names of software producers and the type of programs already being developed. Apple asked us not to release the names of any of these software developers (we suppose to avoid putting unnecessary time pressures on these developers) so we won’t, but we can say that there is quite a list of games being developed. There were also educational programs, business programs, and a large number of specialty programs. The list is impressive and Apple tells us the list grows by about one new program every working day. If that rate continues, it means that Apple will be releasing hundreds of new programs within the next several months. In fact, it appears as though the software developers are getting behind the Macintosh in record numbers. All this should be good news for those considering buying the Mac.

Here are some general areas of computer usage.

Games

If you are buying a small computer and you want to play complicated computer games with it, then you should buy the one that also sells a big stack of tricky computer game programs along with the computer. You might think that would leave out the Macintosh. Generally, it is the low-priced home computers that come with a large variety of computer games. Still, the Mac, with its unique user interface and its very high resolution screen graphics, would make a great games machine, and apparently quite a few games makers have noticed that. Apple has designed its computer with a useful way to manage, display, and print data and graphics. The Mac does these tasks so well that it seems likely that some software vendors will try to adapt those capabilities for the games market. We anticipate that a base of games programs will be created.

Business

What about all those new business planning programs—spreadsheets, memo makers, and database managers? These days, complicated financial forecasting can be done with a combination of spreadsheet analyses and data base organization programs. And Apple is working with many business software producers to make sure that there will be a complete software base for these types of programs. The first thing to look for in a new computer is whether it will run some of the spreadsheet programs. No real problems there; spreadsheet business planning programs are so essential to selling a business computer that just about all of the computer makers include that kind of software in the package. The more specialized, recently developed business planning packages are another matter: Most of these were originaly written for the IBM PC and only now are being converted to run on a variety of other computers. When we looked at Apple’s list of software being developed for the Macintosh, we noted that several of the program packages on the list are the new integrated type of programs. These programs combine many useful programs into one.
For example, you can get one program that combines word processing, spreadsheet planning, and database management.

**Scientific/Math/Accounting**

You say you need to do a lot of number crunching and scientific programming? This can present a problem: Most of this kind of software (what little there is available) was originally written for the IBM PC. There is less of this software available than you might think, given the scientific nature of computers themselves. Scientists and mathematicians usually write their own programs. The experimental nature of their tasks often require it. Still, there are a number of such programs, created for specific tasks, that have been modified for more general use and that are now sold commercially. We are told that several of them will be rewritten for use on the Macintosh.

**Education**

Available software is a key consideration if you are looking at computers for use in education. If you plan to use your computer to run educational programs, to teach yourself and your kids, or if you are a professional educator and you plan to use it at school, you should look closely at the Macintosh. Unfortunately, not many of the small computers on the market today will run much educational software. Education has long been the stronghold of Apple and Atari, but, thus far, little of the educational software made originally for the Apple I, II, and III series computers has been converted to run on the Mac. But we believe there will soon be rapid influx of educational programs made for the Macintosh. Three main considerations lead us to this conclusion: 1) Apple has long been a major force in educational computing; 2) Apple is putting most of its development money into the Macintosh—it is staking its future on this computer; 3) Apple has created and supported a university consortium program. These considerations indicate that Apple is sure to try to maintain its lead in the educational computing market by directing its marketing of the Mac toward schools and home learning. The last factor, the existence of the university consortium program and Apple's continuing program of providing quality discounts and support of other university computing programs, seems to indicate a long-term educational computing strategy.

At the University of Utah (one of the consortium schools) where we teach, there is considerable attention being paid to the capabilities of the Macintosh in all of the educational computing courses. Despite the relative lack of educational programs for the Mac, it is a major focus of discussion whenever the future of computing is discussed. Undoubtedly, the effect of having the Macintosh available in large numbers in the universities was not overlooked as an advantage to Apple’s eventual sales to primary and secondary schools. Because of the large base of software available right now for the Apple II series of computers, our education software evaluation classes tend to focus on those computers. But as educational programs become available for the Macintosh, they seem to attract the lion’s share of in-class attention. We anticipate that teachers trained in these classes will
Buying a Small Computer

carry that interest in the Mac out with them as they move into teaching assignments in the public schools.

The two keys to buying should be starting to become clear:

1) If you expect to use your personal computer for a limited set of applications, shop around for a low-priced computer that will run the few programs you are interested in. Many computers on the market today can be used to carry out a limited number of applications and will do an adequate job for a relatively low price. Just look for one that offers you the type of software that will do your tasks. If you like the hardware (the computer itself), the price/features deal it offers, and the software you believe you will need, buy it.

2) But, if you have some esoteric or highly specialized uses in mind, better lean toward a full-capability computer like the Macintosh. The price/capability quotient of the Mac makes it a good buy for anyone who expects to be using a variety of programs. To this second choice, we might add a note. If you are a "software shopper," if you like to buy and try a variety of software (maybe you are like us and just enjoy learning more about computers and computer programs) then you'd better be leaning toward choice two also. There is a growing catalog of commercial and public domain programs available for the Macintosh, enough to keep a true software connoisseur busy for a long time.

Both of these directions are based on the "buy the software" advice. In a nutshell, our advice is: If you can't get the software you want, don't be tempted to buy the hardware, no matter how interesting it may seem.

HARDWARE CONSIDERATIONS

But there is an aspect of the "Which one is best?" question that does concern hardware—the computer's central processing system. The available software for any computer is always going to be tied to which central processing system the computer uses. This is one of the key issues that affects the potential for a new "Macintosh standard." One of the main reasons the IBM standard came about and resulted in such a large base of applications software was the central processor IBM chose to put into its PC. The IBM PC uses the Intel Corporation's 8088 microprocessor as the heart of the computer's central processing system. At the time the PC was released, the 8088 was an advanced processor, but it did not require any radically new design that might cause problems for the majority of the existing software producers. It was based on the earlier (and very popular) 8008 microprocessor (also made by Intel) and much of the software that existed for earlier computers could be easily modified to run on the 8088 based PC.

But the Motorola 68000, on which the Macintosh's central processing system is based, offers no such advantage. In choosing the 68000 system, Apple has taken a giant step forward in technology, but that step forward is at the expense of compatibility with its existing computers. Apple, of course, was well aware of this when it made the early design decisions for the Macintosh. Apple's decision to move to a more modern central processing system was no doubt well thought out; in fact, it was probably much agonized over since it represents such a large risk to the company itself. The move to the 68000 demonstrates a desire by Apple to incorporate the latest advantages in computer capabil-
ity. Apple is betting that the world is ready for such new technologies and that the added capability will force a significant percentage of computer buyers to abandon the 1970’s technology of the IBM PC.

Let’s look more closely at the implications of computer hardware design, and specifically at the relevance of the central processing system. The Macintosh is not the first computer to utilize one of the hot new 32-bit central processing units.

**Definition: Bit**

A bit is a signal used by computers that is either on or off. It works a lot like a switch. Computer makers design computers to recognize data in one of two states, on or off; they specify these states as either one or zero. A computer uses long strings of these on or off signals as its internal code. The 8-bit size of a block of data became the early standard because most computers used 8 bits to handle one character, such as one of the letters or numbers that you type in from the keyboard. Each letter can be represented by a particular arrangement of these on or off signals.

In fact, several other computer makers have used the exact same microprocessor, the Motorola 68000, in their computers. However, most of these computers were designed to be used in specialized, often scientific, applications that require high speed mathematical or graphic uses. IBM even makes a specialized scientific lab computer that uses the 68000.

The Intel 8088 used in the IBM PC and its compatibles internally handles data in 16-bit chunks and handles its input/output (I/O) in 8-bit chunks. That’s why the Intel 8088 is referred to as an 8/16 bit CPU. Some of the newer and faster CPUs can handle data in up to 32-bit chunks. This gives them more capability and makes them operate much faster than true IBM compatibles. The Macintosh is sometimes referred to as a 32-bit computer because the 68000 processor is capable of handling internal data in 32-bit chunks; but it more properly should be referred to as a 16/32-bit computer because it handles data input and output in 16-bit chunks. Even so, this capability makes the Macintosh a much faster data handler than the IBM PC; faster, in fact, than most of the computers on the market today. The method of data handling is built into the CPU. Some of the newer computers use advanced central processing units somewhat similar to the 68000 such as the Intel 80186 or 80286 16/16-bit CPUs.

All these CPU names and bit-handling capabilities may seem pointless, especially if you don’t know much about how small computers work in the first place. But if you pay attention to happenings in the computer world, you will soon be hearing more about these new processors. The debate over their usefulness is just beginning. At issue is whether the computing public really needs the increased speed and capability of these new processors. Already, the battle is joined by the Macintosh and Lisa; their introduction demonstrates Apple’s willingness to break away from the IBM standard in order to use the increased capability of the new Motorola 68000 central processor. By making this choice, Apple
accepted the fact that its new computer could run none of the IBM-standard software. It meant that its new computer entered the market with very little software support, at least compared with the host of programs available to producers of IBM-compatible computers. Still, there was little software available for the IBM PC when it was first introduced. Apple is counting on its well-known name in the consumer field and, especially, on its established presence in the educational market, to encourage the production of new software for the Mac. Whether this move will lead to the creation of a second standard in the small computer industry, the Macintosh standard, still remains to be seen. As we pointed out in Chapter 1, Apple’s decision to move to the more advanced central processor also means that none of the software created for the Apple I, II, or III series computers will run on the Macintosh. Not only are the central processors different (the older machines use the MOS 6502 processors that handle data in 8-bit chunks), but the Macintosh uses a smaller, 3½ inch high-density diskette.

To give you a clearer idea of the latest developments in CPU design, we have included a chart listing the central processors currently in common use in small computers and those likely to be used in the future.

<table>
<thead>
<tr>
<th>MODEL NAME</th>
<th>MANUFACTURER</th>
<th>I/O DATA HANDLING</th>
<th>INTERNAL DATA HANDLING</th>
</tr>
</thead>
<tbody>
<tr>
<td>8008</td>
<td>Intel</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>8080A</td>
<td>Intel</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>8085</td>
<td>Intel</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>6502</td>
<td>MOS Technology</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Z80</td>
<td>Zilog</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

8-Bit Chips (1970s)

<table>
<thead>
<tr>
<th>MODEL NAME</th>
<th>MANUFACTURER</th>
<th>I/O DATA HANDLING</th>
<th>INTERNAL DATA HANDLING</th>
</tr>
</thead>
<tbody>
<tr>
<td>8088</td>
<td>Intel</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>8086</td>
<td>Intel</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>80186</td>
<td>Intel</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>80286</td>
<td>Intel</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Z8000</td>
<td>Zilog</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

16-Bit Chips (Early 1980s)

<table>
<thead>
<tr>
<th>MODEL NAME</th>
<th>MANUFACTURER</th>
<th>I/O DATA HANDLING</th>
<th>INTERNAL DATA HANDLING</th>
</tr>
</thead>
<tbody>
<tr>
<td>68000</td>
<td>Motorola</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>68010</td>
<td>Motorola</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>NS16016</td>
<td>Nat. Semiconductor</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>NS16032</td>
<td>Nat. Semiconductor</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>WE32000</td>
<td>AT&amp;T Bell Labs</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>HP FOCUS</td>
<td>Hewlett-Packard</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>80386</td>
<td>Intel</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>MC68020</td>
<td>Motorola</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>NS32032</td>
<td>Nat. Semiconductor</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>NCR/32</td>
<td>NCR</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Z80000</td>
<td>Zilog</td>
<td>32</td>
<td>32</td>
</tr>
</tbody>
</table>

32-Bit Chips (1980s-future)
The Speed Question

The processor chart demonstrates some of the changes that have taken place in CPU technology in the last few years. Years ago, the manufacturing of microprocessing chips was not all that profitable. There just weren't that many small computers on the market. But now that microcomputers are a multibillion dollar business, there is a lot more competition. Among the early entries were the Intel 8008 CPUs, introduced in the early 1970s as an 8-bit upgrade to Intel's 4004 CPUs, which were 4-bit units. The Zilog Corporation also got in on the early boom of small computers in the 70s with its very popular Z80 CPU chip. MOS Technology came out with the MOS6502 processor at about the same time and it soon became number one as both Apple and Commodore came out with small computers based on it.

The present generation of chips is mostly 16-bit, moving toward the 32-bit units. And there is no reason to believe the technology will not continue to change. There is always a slightly delayed public response to new technologies—we are usually satisfied (for a while) with what we have. But sooner or later we see somebody down the block with the latest thing, and it is not long before we want it too. Right now, the newest thing down the block is the Motorola 68000 series of processors. The Apple Macintosh uses one of them and since it is one of the fastest selling computers on the market, it is sure to make the capabilities of the CPU well known.

On the horizon is Intel's 80000 series processors, already gaining wide acceptance. In many applications, these new processors operate at much greater data-handling speeds and give the user access to a much larger range of memory addresses than the 8086 of the earlier IBM PCs and its compatibles, which are restricted to 640K bytes of user memory. And there are even newer and faster processors still to come. In computer circles, the NCR 32-bit CPUs and the 32-bit entries from Zilog and National Semiconductor are much talked about. Who can say which will be the star of the future? The ones that succeed in attracting the interest of the third party software producers will undoubtedly be the winners. The rest will fall by the wayside.

THE SPEED QUESTION

All the ads say that their computer is faster than anybody else's. Does that really matter?

Answer

The question of speed is important, but only up to a point. The most common use of a computer will be to run the applications programs you can buy for it. Your computer needs to be able to run those programs at least as fast as the program designer wanted them to run. Let's say you buy a program called "Little Johnny One Note" that is supposed to teach kids how to read music. If Little Johnny One Note is jogging across the computer's screen too slowly to maintain the program's effect, then the computer is killing the program. The kids you are trying to teach are likely to be confused or lose interest. The problem is that the program does not run as fast as was intended. That is the key concept; how fast is fast enough?
Does the computer run games and educational programs as fast as the programmer intended? What is a slow computer and what is a fast one? These are questions that even the experts have trouble finding answers for.

But if speed is your concern, you needn’t worry. The Macintosh is one of the fastest computers on the market. Because of its 16/32-bit data-handling capabilities, it can outrun almost any other computer now available.

The biggest speed problem a user has to deal with is "wait time." Wait time is the time you spend waiting for the computer to finish up its internal housekeeping before you can go on working. A computer can seem fast enough when you try it out in the store, but once you get good at using it, long wait times will be a constant annoyance. It may not be reason enough to toss your new computer into the garbage can and go back to the store to buy a different one, but while you are waiting for a slow computer to get its tasks done you sometimes think strongly about giving it the boot.

The main indication of a computer’s speed (and therefore its wait time) is how it handles data; that is, how it moves data around inside the computer. A lot of home computers move data in chunks that are 8 bits long. As indicated in our microprocessor chart in the last section, a few years ago 8-bit processing was the standard.

As we pointed out earlier in this chapter, 8-bit computers can only handle 8 bits of data at a time. Eight-bit computers maintain this 8-bit consistency even in their internal handling of data. These 8-bit computers were limited by how fast they could handle data in little 8-bit pieces, so the newer 16-bit computers were designed to handle data in 16-bit chunks. If you do some very simple arithmetic, you will almost get the right answer: Since a 16-bit computer can handle chunks of data that are twice as big as the chunks the old 8-bit computers handled, they will run twice as fast. Well, almost. The actual operating speed is limited by such things as diskette access time and screen display handling, but a good 16-bit computer can run almost twice as fast as the old 8-bit computers. The Macintosh handles its input and output of data in these faster 16-bit chunks. And, just as significantly, it can handle data in 32-bit chunks in all its internal operations.

One important data-handling task is how the computer talks to its internal memory storage areas. The main storage area in a computer is called random access memory, or RAM for short. The inner workings of computer interaction with RAM are quite complicated and a lot of the computer’s time can be used up just dealing with it. When you use a big, complicated program, say a word processing program, the computer first stores all of the program’s instructions in RAM. Then, whenever it receives text typed in from the keyboard, it has to go back to look in its RAM for word processing instructions about how to handle the text. The wait time is directly related to how long the computer takes to do such internal housekeeping tasks.

So, back to our original question: Is the Macintosh a fast computer, or is it a slow one? The answer is: Well, it depends. It depends on what you are comparing them with. What we mean is that you should keep in mind the fact that there are relative speeds. A computer might be fast at adding up a long list of numbers, but slow at running an applications package. Or it might be fast at running one type of word processing program but slow at running other types of programs. It depends, to some degree, on how the computer is designed and how the particular applications program interacts with it. The best way to tell is to compare computers using benchmarks. Benchmarks are programs that
cause the computer to act in predictable ways. Once you have a program that is predicta-
ble, you can observe how it acts while the program is running and then compare that to
how another computer acted while running the same program. Timing how long the pro-
gram takes to run on several different computers is one common benchmarking method.
Other standard programs test what a computer does when overloaded, or how it acts when
given an unusual task. Does the computer fail or does it have a built-in way to recover and
keep going? If the benchmark program forces the computer into an error condition, does it
present the user with a helpful error message and continue, or does it just freeze up and
refuse to do anything at all? These are all different forms of benchmarking, and they are
among the ways technicians learn about how a computer is designed. Not much of this
type of information would be of help to the average person getting ready to buy a
computer—it is just too technical—but the benchmark tests of speed are easy to under-
stand and may give you a good idea of how useful the computer will be to you.
Benchmark performances for the various compatibles are often published in the popular
computer magazines. And a dealer selling any one type of computer should also be able to
give you benchmark results for that computer. Any seller of computers should at least be
able to give you standard speed comparisons with the IBM PC and a few of the other
popular personal computers. Make sure they give you real facts and figures, rather than
vague promises. Some promise that their computer will run “twice as fast” as the IBM
PC. You can ask them for more complete details, such as on what type of programs was
this speed test done? Was it fast at internal memory accessing tasks? Or fast at accessing
files on a diskette? How fast is it when controlling a printer attached to the computer?
Computer dealers like to brag about their computer’s speed. Since they brought up the
subject, make them prove it.

And, if you are considering the Macintosh, you should be aware that it has an ad-
vantaje in the speed category. As we pointed out above, the Motorola 68000 processor
used in the Macintosh is much faster than most of the processors used in today’s comput-
ers. This should make the Mac much faster than almost anything you can find to compare
it with.

THE MEMORY QUESTION

Some of the ads on TV say their computer is the best because it has the most memory.
What kind of memory are they talking about? How important is it to have a lot of
memory?

Answer

The answer to these questions about memory is a lot like the answer to the questions about
speed: It is important to have enough of it. And how much is enough? That’s up to you,
because that is closely related to how you plan to use the computer.

If you want to play computer games with your computer, you only need enough
memory to load and store the game programs. But if you want to run one of the new
windowing programs, one of the powerful programs that allows you to have several pro-
grams going all at once with some part of each of them continually displayed on the screen (usually kept separate in boxed-off areas called windows), then you will need a lot more memory.

The original IBM PC came with 64,000 bytes (abbreviated 64KB) of random access memory (RAM). Some other popular computers, like the Apple II and the Ataris and the home computers from Radio Shack came out with a lot less than that. In the early years of computers, 64KB would have been thought of as a tremendous amount of memory. When the IBM PC was first announced a few years ago, 64K bytes of RAM was about the standard. IBM could have included more—the technology was available—but RAM chips were still fairly expensive and IBM didn’t want its first entry into the small computer market to look too expensive. The fact is that 64KB is not really enough memory for most of the new computer applications. The new word processing programs are a good example of what we mean. Word processors have become a lot easier to use in the past few years, and in the process they eat up a lot more memory. Making things easier for users can mean sorting a lot of help messages. Many modern word processors include quite a lengthy file of helpful instructions on how to use some of the program’s advanced features. These help files are stored in user RAM (sometimes called user memory) as soon as you load the program. Add this to the room already used up by the computer’s main operating instructions, and put that on top of the word processing program’s operating instructions, and you can see how quickly you can run out of room.

Some of today’s word processing programs require your computer to have at least 128KB of user memory to operate, and some of the newer integrated programs require twice that much. The problem is that there has to be enough room in RAM to store the operating systems, the help instructions, and still have room for all those words you are typing in. Before long, the computer will start running out of room in memory, causing the program either to start giving you error messages or else start temporarily storing new text entries on the diskette. But if the program has to use diskette storage, it will eventually slow down so much as to be unusable.

**Definition: RAM and ROM**

RAM stands for random access memory. It doesn’t mean that you access whenever you feel like it, but rather that you can access at any specified address. But the key thing to remember about RAM is that it is a different kind of memory than ROM (read-only memory). ROM can also be accessed at any specified address, but that access can only be to read from it. Compare this to RAM which can be read from, like ROM, but also be written to; in other words, you can add things to be stored in RAM at any time (as long as it is not filled up with data). That is why it’s known as user memory. ROM is usually used for instructions to the computer and it is so important to the operation of the computer that the ROM chips are sometimes called firmware, a distinction that puts it somewhere between the actual wires, printed circuits, and diskette drives (the hardware) and the computer programs (the software).
Today, most computers come with at least 128K bytes of RAM. Even though this is twice what the older models usually came with, we were told by one computer manufacturer that few users today are satisfied even with the 128KB. They apparently are buying add-on memory boards faster than they can be built. What is happening is that the new multipurpose software packages are designed to be able to run on 128K, but barely. Once customers buy this powerful new type of software they are generally happy with what the program does, but they are unhappy with how slowly it does it. The problem has to do with the contrast between RAM access time and diskette access time.

If there is not enough internal memory, the program will start storing the overflow on the diskette, just as we pointed out in our description of how word processors work. In order to retrieve something from active memory (another of the many names for RAM) and display it on the screen, the system needs only ask for it. Data can be retrieved from RAM and displayed very quickly. It is available faster than your finger can press down the key that asked for it. Compare that with how long it takes to retrieve information from a diskette. In order to do that, the operating system has to send a request to locate a specified address on the diskette. Once that address is determined, the diskette drive motor has to be activated, the read/write head has to be moved into position to hover above that diskette location, the information has to be read back into the computer, and then, finally, it can be displayed on the screen. It’s no wonder that 256K bytes of user memory is fast becoming the standard for personal computers.

And so when users complain about how slow their programs are running, everyone advises them to go out and get some more memory; they are told their programs will speed up dramatically. So eventually they buy the board, and guess what? The program does speed up. This all brings us back to one of our original questions, how much memory is enough? One answer is: enough so that lack of internal memory isn’t slowing down your program.

And does the Macintosh have enough? It has plenty for today’s programs. You can get anywhere from 128K bytes up to 512K bytes. We recommend the 512K systems for those planning to use a variety of software. It not only assures you of enough memory to run almost any type of program, but will also speed up the operation of the computer.

A word of caution here when you are considering buying add-on memory: Make sure, before you buy add-on memory for your computer that it will work correctly. First, make sure the memory chips were designed for your computer. Most of them will work as advertised. The problem is, many of the commercial applications programs (especially the ones written in assembler language) do error checking routines that check to see if a programmed task is completed. These programs will not proceed to the next instruction unless they get a satisfactory report on each error check. Since some of these checks are timing dependent, slow memory handling (when the added memory on your new memory board is called on) can result in error. The next thing you see is the dead stop of instruction processing. The program stops because it doesn’t know what to do next. To you, the user, this may mean the keyboard will lock up and accept no more entries. The only way to get going again is to turn the computer off or to reset it. Either way, you will have lost any data that was not saved to diskette. There is no worse feeling than to lose all those precious thoughts that you had just typed in, but had not yet saved on the diskette. This kind of problem can be very troublesome as it can happen inconsistently and without
warning. You might never suspect that it is the memory that is causing the trouble. There are two things you can do to protect yourself: If the memory was not made for your computer, try to find out if it has also been tested on the computer you are buying it for. Most add-on memory manufacturers will gladly provide that kind of information. If possible, get the specs on how fast the memory board chips are accessed. This will be a number measured in nanoseconds. A board with a rating that is 2 nanoseconds slower than your program expects to find can cause you a variety of problems. Shop around. Buying the one with the fastest rating is usually best, although it may cost you a little more. The second way to protect yourself is by making sure you can bring the board back if you are dissatisfied with its performance. It is a good idea to get this kind of a guarantee with any part you buy for a computer. The reliable manufacturers will always offer such a guarantee.

Often, memory size is about the only real (factual) information the computer ads will tell you. But what does it mean, and how valuable is having a lot of memory? Internal memory is one of the keys to modern computer usage. The basic idea is you can store data in active memory and you can ask the computer to access it at any time (as long as either you or the computer knows where to find it). Sometimes the ads also tell how much read-only memory (ROM) the computer has. They make it sound as if the more ROM you have, the better your computer is. But the truth is that ROM is a storage area used only by the computer. The key question to ask is, what is stored in all that ROM? You, the user, can't use it. It can't be used to store your programs, not even temporarily. It is where the basic instructions, like how to display letters on the screen, or how to send information back and forth between different parts of the computer, are stored. The instructions permanently stored in ROM are called firmware. A programmer can look at what is in ROM (read it) but he can't write anything in there. That's why it's called read-only memory.

External storage is different. External storage is usually not called memory. The most common type of external storage used on the IBM-compatible computers is the standard 5¼-inch diskettes, sometimes called "floppy disks" to distinguish them from the much higher capacity hard disks (sometimes called fixed disks).

But the Macintosh uses a new type of diskette, smaller (3½-inch diameter) that, despite its size, holds more than the 5¼-inch floppies. Read more about these new diskettes in the chapter that takes a closer look at the hardware.

THE MONEY QUESTION

At about $2200, buying a Macintosh is not a decision to be lightly regarded. Wouldn’t it be smarter just to buy one of the cheap home computers?

Answer

No. It is not best to just buy the cheapest computer you can find because it may not do the job for you.

But that’s just the short answer. Computer selling at the low end—the cheap, home-
Firmware is a term that refers to instructions to the computer that have been permanently burned into the ROM (read-only memory). As we said, the name comes from the fact that firmware occupies somewhat of a gray area between a computer's hardware (the physical parts) and the software that is loaded into the computer to make it perform. Firmware instructions are programs, like software, but the difference is that the code is permanently available to the system whenever it is needed. Instructions to the computer that do not change with each application—such as how to address the diskette drives or how to load a program into memory—are typical of the firmware instructions stored in a computer's ROM.

The total set of the instructions that are used specifically to control the basic functioning of the computer is often referred to as the BIOS (the basic input/output system). The ROM can also include entire programs that are permanently stored there. BASIC is an example of this type of permanently stored program. Or the ROM may hold help messages for users or even a variety of type fonts and special characters for users who want to be able to type in foreign language letters or numbers. Some of the portables are also beginning to offer spreadsheet programs and word processors, permanently available from ROM. This is an idea that may soon be picked up by other manufacturers to make their computer stand out from the crowd.
The overall answer is that there is no easy answer. You have to know what you want to do with a computer so you can buy the system that has the capability to do it. And what if you don't know, in advance, what you are going to use it for? Well, then maybe the best answer is to get the one that is general purpose, the one that runs most of the available software, one that can be expanded later to do more tasks.

Is the Macintosh that sort of computer? We think it is. It is easier to use, faster, and comes with plenty of memory. And despite its reputation as a beginner's computer, it will actually do anything any other computer will do, and usually it will do it better. It should not be compared with lower-cost computers, not even the ones made by Apple. This is a new-age computer, one that incorporates a new, easier-to-use software design with the fastest, most flexible hardware systems on the market. Add to that its graphics capability and ability to print high-quality hard copy, and it's easy to see why the Mac is selling at a faster rate than any mid-priced personal computer in history.
In this chapter, we take a closer look at the parts and pieces that make up the Macintosh computer. In the computer world, a computer system is said to be made up of two main components, the hardware (the actual computer), and the programs that make it operate (the software). In a computer as complicated in design as the Macintosh, we feel it is important to analyze in more detail those two main components. In this chapter we will look at each of the hardware components that make up the Macintosh system, and in the next chapter we look at the second component, the programs that comprise the Mac's operating software.

PARTS AND PIECES

When referring to computer components it is proper to call them "hardware." Strictly speaking, any part of a computer that is a physical part of the machine is part of the hardware. But, somehow, calling the Macintosh just a piece of computing hardware seems, well, rather sacriligious. There is a mystique to the little computer from Apple, or so it seems to the true believers. We refer to them as MacUsers. They see the actual electronics of the Macintosh as merely a medium within which to carry forth a new message: COMPUTING CAN BE FUN! Or, COMPUTING CAN BE EASY! To us, computing has always been fun; we like to play with computers, to put them through their paces, to find
out what makes them tick. But the Mac brings with it a new kind of fun, a new user "interface" (a computerese word that refers to the point at which user and computer come together, something like "interaction").

To keep the dedicated MacUsers happy, we will call this section "Parts and Pieces." It's really about more than just the hardware anyhow, because where the Macintosh really demonstrates its innovative character is in the interaction between hardware and software. It's a carefully integrated machine that ties a well-thought-out hardware design to a remarkable set of easy-to-use operating programs.

The Macintosh was designed, above all, to be simple. As a result, the parts of the computer are, for the most part, simple and easy to understand. The keyboard is small and unadorned. It has fewer keys on it than almost any computer on the market. It uses a mouse-type pointing device that has just one button and is easy to use. But there is a paradox to all this simplicity. In order to make the computer easy to use, the processing design of the computer had to be very complex. In order to make the machine easy to understand, the user has to be "protected" from much of its capability; that is, the operating system gives the user easy access to the computer's capability, but the computer has to work harder to do it.

In order to understand how the system that is the Macintosh fits together, let's look at each of its components.

**THE KEYBOARD VERSUS THE MOUSE**

Most computer users are not concerned with what electronics are inside the box. Most people don't care how a computer works, they just want it to do the job. Teaching both adults and kids to use computers, we have learned that they think of the computer not as made up of wires and integrated circuits, but as a box with a keyboard and a screen. The point of interface is usually at the keyboard, where they type data into the computer—then the results of that action appear on the screen. While this may seem self-evident, what is not so obvious is that because of this interface it is the computer's keyboard that often determines how users see the entire machine. Apple took this into account when designing the small, easy-to-use keyboard. A big, complicated keyboard with lots of special function keys can be helpful to an experienced user, but to a less experienced user it can make the whole computer seem overly technical and hard to operate.

Computers that came before the Macintosh tended to expand their computer keyboards until there was no room left for any more keys. Then they started adding keys that carried out special functions. The idea was to make it easier to use the computer by keeping the keystrokes down to as few as possible. But Apple took this concept one step further by trying to minimize the number of things you had to do in order to use the computer.

When Apple started to develop the Mac, it asked the question: Instead of pressing keys to make things happen, is there a better way? Typing in commands on a computer keyboard means learning exact names and syntaxes. Figuring out what command needs to be typed in to do which task requires learning quite a lot about how the computer and its operating system works. The command names were typically short, coded syllables that
often stood for computer functions rather than for computer tasks. While it may have been obvious to the typical programmer that the command

\[
\text{PRINT [D:]PATH \{[D:]\ PATH} \ldots \})
\]

could be used to queue up a set of data files to print, imagine the poor inexperienced user trying to figure out how to use the following commands:

\[
\begin{align*}
\text{CHKDSK [D:] [FILENAME[.EXT]] [F]/[V]} \\
\text{FIND [/V][/C][/N] STRING[\{D:]\{PATH\} FILENAME[EXT] \ldots]}
\end{align*}
\]

\[
\text{TYPE [D:]\{PATH\}FILENAME[.EXT]}
\]

These three commands are used fairly commonly on all IBM-compatible personal computers. Even experienced users often have to go scurrying for the operating guide to remember how to use such commands.

Part of the reason computers are notoriously hard to learn to operate, is because of these function-oriented command sets, which require a lot of knowledge about the computer before you can even begin to go to work.

The designers at Apple probably thought, what if you didn’t need to type in a command? What if there was a list of the things the computer could do? Then you would only have to pick the function you wanted from the list. And once the designers had made that decision, the next step became obvious: If you don’t need to key in commands, why do you need the keyboard? Is there a better way to point to one choice on a menu of choices. The “mouse” type of pointing device eventually evolved as the best method.

**THE MOUSE**

For those not familiar with the mouse, it is simply a small box, about the size of a pack of cigarettes, that is connected by wire to the back of the Macintosh. The picture of the mouse and the keyboard in this section shows the relative size of each. The keyboard is about 13 inches long and 5½ inches wide. The mouse is about 3½ inches by 2½ inches. Since the Macintosh popularized the concept, mechanical “mice” have started to catch on with other computer makers. The technology is not very complicated; the idea itself was not even new. It just took a well-designed, relatively inexpensive computer to bring the idea to the attention of most users. But while some of the other computer manufacturers are making two button or even three button mice, the Macintosh mouse has only one button on top. You roll the mouse across the table to move the pointer to the position you want on the screen. Then you press the button. Simple. Like the design of the keyboard, Apple decided to make the mouse simple and easy to use. All Apple’s design decisions followed this strategy.

Apple reportedly tried a lot of other pointing and selecting devices before settling on the mouse. It supposedly examined the feasibility of a graphics board input (you write on a graphics board with a wooden “pencil”), track balls, and light pens, as well as the more familiar joysticks and cursor keys. Those who like to keep their hands on the keyboard,
and those who have grown used to other computers, may have trouble getting used to reaching over to grab the mouse every time they want to move to a different point on the screen. For them, Apple might have found room on the keyboard for the traditional cursor keys (also called arrow keys). For those who have grown used to very messy desks (as we must confess we have), it might not be an easy task to find enough room on the desk to manipulate the mouse. Given the choice, we messy desk types might have chosen the track ball; it can do the same tasks as the mouse, but can be mounted on the keyboard. We have even seen people move the screen pointer by picking up the mouse and spinning the ball buried in its base.

Still, the research people at Xerox's Palo Alto research labs decided the mouse was the best pointing device too—long before Apple started to develop the Macintosh. So who are we to argue? If you use the mouse faithfully (not depending too much on the combinations of keystrokes that can be used to do the mouse's job), you will soon get used to it and be able to fly the pointer around the screen and land on those little 'click boxes' like the experts. By the way, most of the real mouse experts we have seen are kids who were challenged by the mouse and have learned to manipulate it with all the skillful eye-hand coordination they had developed after endless hours at the joystick controls down at the video arcade. They can double click and drag boxes with the accuracy and speed of an Outer Galivanian Starfighter Pilot.

Like the Macintosh itself, probably the most interesting thing about the mouse is the software it takes to support it. All that clicking and double clicking, the screen display that shows the pointer moving across the screen no matter how fast you zip it, those multiple keystroke combinations that can be combined with mouse manipulation to produce special effects in MacPaint and in other programs—all that takes a tremendously complicated
operating program, combined with very powerful and fast processing hardware. Think about one apparently very simple effect. When you move the mouse pointer to one of the key areas on the screen (a click box, a menu label, a dragging area of a box) and click the button, the software and the processing system is smart enough to detect where the pointer is and to take the expected action. But move that pointer to any other point on the screen and click the mouse button. What happens? Nothing! This may seem simple: There was not a pointer match at a critical spot on the screen. But it means the processing system has to almost instantaneously determine the pointer position and then scan the screen for one of the many different types of critical points every time the mouse button is pushed. It has to make the correct response every time you press the button, and even if it finds no match, it has to interrupt what it was doing to conduct the test. Yet no matter how many times you do it, you probably won’t be able to detect any slowdown in your programs. The Mac software is that sophisticated and the processor is that fast.

Another interesting thing about the mouse-support software is that the onscreen pointer sometimes changes shapes. It can be an arrow, a crosshair (as in a rifle scope, to indicate a junction), or a vertical line (to indicate the insertion point for text). The size and shape of the pointer is up to the programmer. Pointers in Mac programs have been seen in the shape of pointing hands, dancers, space invaders, paint sprayers, and a variety of other odd and unnamed shapes. The pointer will often even turn into a little wrist watch to tell you to wait a bit. Think of the complexity of the computer program it takes to turn the pointer into a watch, and of the software support that allows you to still continue to move that watch/pointer so you can get it to the right place it needs to be on the screen when it turns back into an arrow.

MOUSE FUNCTIONS

There are a limited number of actions you have to learn in order to use the mouse. They are described in the chapter on the Macintosh’s operating system, but we thought it would be useful to summarize them here too.

Click

This is Apple’s term for pressing the button that is on top of the mouse. You are supposed to press it down quickly and release it. Clicking is used for one purpose, to select a file or an operation. When you select a program file, the screen representation of that file (the icon) will often be highlighted by reverse video. That means that the icon and the box that surrounds it will be shown with reversed light and dark areas.

Double Click

This is the same as a click, except you press the button twice. Be sure to press the button twice very quickly, to be sure the computer doesn’t interpret your signal as two single clicks. Double clicking is usually used to select a function or a program file, but it can also be used for any number of other applications. For example, in the MacPaint program, all your completed drawings show up as files after you have saved them on the diskette. And each has its own pictorial icon representation. To tell the computer to display one of these files, just move the mouse pointer to the picture you want and double click.
Drag

This function with the funny sounding name is Apple's term for selecting by pressing the mouse button and then holding it down while the mouse is rolled across the desk top. You move the mouse pointer to a special spot on the screen, select the related function by pressing the mouse button, and then move the mouse. When you release the button, the dragging action stops. So of what use is dragging? There are actually as many uses of dragging as a programmer might want to make up for it. One use is to select an item from a menu. You can move the mouse screen pointer to the menu-select line at the top left edge of the screen and press the mouse button. This activates that menu, and, when you drag the mouse down (still holding down the button), the menu is displayed. Dragging the mouse down a little further will highlight the first menu selection. Continue dragging and the other items will be highlighted until you run off the bottom edge of the menu. If you release the mouse at any point in the dragging process, the item that was highlighted at that moment will be activated. Again, like many of the other features of the Macintosh operating system, this is an impressive feat of computer programming. With earlier computers, you would have had to either type in the name of the menu selection or at least type in a letter or number corresponding to the item on the menu. Another use of dragging is to select. You can drag the mouse across one or more icons that represent program files and they will all be selected.

Shift Click

Shift clicking means to press the keyboard shift key and hold it down while you press the mouse button. By adding the input of the shift key, programmers can create more complicated uses for the mouse. Clicking the mouse while pressing the shift key is used for special applications in the MacPaint, MacWrite, and other programs designed especially for the Macintosh. Shift clicking is just one more way software designers can use the mouse. In fact, the mouse functions can be combined with any other keystroke or combinations of keystrokes to perform almost any software-designed role.

THE KEYBOARD

Once the decision was made to use a mouse for pointing out choices on the computer's screen, what role was left for the keyboard? Answer: for typing. The Macintosh keyboard has the same purpose as a traditional typewriter keyboard. In fact, the Macintosh has only about the same number of keys as an IBM Selectric electric typewriter. Apple says you can do a great many things using the Macintosh without ever touching the keyboard. Some software programs, like MacPaint, hardly require the keyboard to be connected. But Apple assumes that you will still be wanting to type some letters and memos. Maybe you will even use the keyboard to program in BASIC. The mouse can be used to point to functions or to move things around on the screen, but to do any typing, to use Apple's text editing program (called, naturally, MacWrite), you will also need the keyboard.

TYPEWRITER KEYS

Most of the keys on the Macintosh keyboard are the same ones (in the same places) as on an ordinary typewriter. There is a Tab key and two Shift keys and a Backspace key, just
like on a typewriter. There is also a Return key, like the one found on electric typewriters; however, this key has some functions on a computer in addition to the carriage return function (refer to the section on the Enter key below).

The Tab key also acts a little differently than on a typewriter. When some menus or “dialog boxes” are on the screen, the Tab key can be used to move quickly through the choices.

SPECIAL KEYS

If we class the typewriter keys on the Macintosh keyboard as one functional group, what are we left with? Only three types of keys. There are two option keys, an enter key, and a strange-looking key that has a symbol on it that looks something like a four-leaf clover (\[\text{ символ }\]). There are only 58 keys on the keyboard. On a typical microcomputer keyboard these days there are likely to be 80 to 100 keys. There would probably be at least ten special function keys, an Escape key, a Control key, an Alt key, Number Lock, Scroll Lock/Break, Print Screen, Home, End, Page Up, Page Down, four arrow keys (called cursor keys) and an extra set of number keys. Sometimes there are even special single function keys, to do things like erase characters or to duplicate lines. Apple apparently believed all these keys were making it tougher to use the computer, not easier.

Apple chose to avoid adding any keys that were not absolutely necessary. If we assume you already know how to use a typewriter keyboard (including Backspace, Shift, Tab, Return, and Caps lock) let’s look at the ones you may not know how to use.

ENTER KEY

This key, like all others, can be put to various uses by different applications programs. Often the Enter key and the Return key are programmed to carry out the same functions. In the programs that were announced along with the Macintosh, it was usually used to respond to an “alert box.” These boxes usually appear on the screen only to notify you that an operation has been completed and/or that another operation is about to begin. You can often move the mouse pointer to a box that says “OK” or “Go Ahead.” The Enter key allows you to do the same thing without taking your hands off the keyboard, and often you can do it more quickly. When using the MacWrite program, you can use the Enter key to open new files, or, in the Multiplan program from Microsoft, Inc., you can use it to enter data into cells.

OPTION KEY

The Option key is used only for creating special symbols. By holding down the Option key while you press one of the other keys, you can create a variety of foreign-language letters and characters. Of course, software producers may find new uses for it; for exam-
ple, in the MacPaint program, the key may be used to make copies of your art. Shown below is a representation of the special symbols that can be produced using the Option key.

A few more special characters can be created by using the Option key and a letter key, after first setting the Caps lock key. They are represented below.

**COMMAND KEY**

The Command key, the one with the four-leaf-clover design on it, has a variety of uses. You can think of it as a shortcut key, used to do things you would usually do with the mouse. The mouse device is a great solution to many of the problems created when a user has to interact with a computer's operating system: It can be used to point or select, thereby making it unnecessary to type in commands. However, all mouse-type pointing devices have one serious problem: You have to take your hand off the keyboard in order to use them. As users become more proficient, it may be more trouble than it's worth for them to stop typing long enough to manipulate the mouse. So Apple has given such users another way. The Command key can be used in conjunction with other keys to mimic some of the functions of the mouse. For example, many of the mouse-controlled editing functions—the kind of functions you would use if you were using the word processing program—can be duplicated using the Command key and one of the letter keys.

Although software can redefine any key to perform a great many different functions, Apple has tried to minimize the uses of the Command key and to keep the key functions consistent across programs. The designers of the Macintosh software have also attempted to use editing Command key functions in the MacPaint and MacWrite programs that are the same as in the operating system program. This should make it easier to memorize them. Apple has suggested some "standard" uses for this key in the programs released with the Mac. Many of the functions are keyed to the first letter of the desired effect: for example, the combination of the Command key and the letter D can be used to duplicate a
file. Other command key combinations, like **Undo**, are placed conveniently close together on the keyboard.

If you can learn to use the **Command** key functions with your left hand while manipulating the mouse with your right, you can modify screen entries very quickly.

**SOFTWARE USAGES**

These special keys have been put to use doing special tasks in the applications programs that were released in the early days of the Macintosh.

The MacPrint program, for example, uses the **Command** key to stretch a selected part of a drawing. You just hold down the **Command** key while dragging the selected part of the drawing with the mouse.

Sometimes the software designers require you to hold down three keys at once. When using the MacPaint program, you can use the **Command** key plus the **Shift** key plus the number 3 key to take a "snapshot" of the screen. When you press these three keys all at once in the MacPrint program, the contents of the screen are saved as a new file. You can also print out to an attached graphics printer whatever is showing on the screen by pressing the **Command** key, the **Shift** key, and the number 4 key.

There is also a special way to make copies of art when using the MacPaint program. To make a single copy, you first select the part you want to copy, then press the **Option** key and use the dragging feature with the mouse. The **Command** key can be used in conjunction with the "pencil" feature by pressing this key and holding it while clicking the mouse button; this will let you automatically change into the FatBits feature of the program.

In other programs, the **Command** key has other functions. For example, when using Microsoft's Multiplan program, pressing the **Command** key while clicking the mouse pointer in a cell will make the cell active.

Microsoft's Word uses a variety of **Command** key options. Usually, the **Command** key is pressed in combination with the **Shift** key and one other key to specify editing options.

**NUMERIC KEYPAD**

Apple also sells an additional numerical keyboard for the Macintosh called the Numeric Keypad. It has 18 keys, including numbers and the **Clear** and **Enter** keys typically found on handheld calculators.

There are also a couple of keys with arrows on them that can be used to move quickly through the numbered cells of planning spreadsheets.

Of course, the keys on the numeric keypad can be programmed by applications developers for other uses (as can any of the other keys on the main keyboard). Future programs might find a variety of uses for the numeric keypad.
DISK DRIVES AND DISKETTES

The Macintosh was one of the first computers introduced with disk drives that could use the smaller-sized, 3 ½-inch hard-cased diskettes. These diskettes are much more durable than the traditional 5 ¼-inch diskettes, which became known as floppies because of the thin plastic case that allowed them to be easily bent. It is this flexible plastic case that can spell doom for a diskette: If you accidentally sit on one, it is probably done for. This is the main reason Apple decided to use the much tougher Sony hard-cased diskettes.

It may be important to pause here and explain exactly what a diskette is. All diskettes—both the old 5 ¼-inch floppies and the new Macintosh 3 ½ inch hard-cased models—are basically made up of two parts. The case is one part, but the important part is the thin disk of magnetic recording medium hidden inside the case. The diskette drive mechanism includes a read/write head that skims along the surface of the recording medium to magnetically encode the data onto the disk. The disk is spun at high speed by the diskette drive motor, and the read/write head works in conjunction with the drive’s known spin speed to magnetically store information on the diskette (known as “writing to the diskette”). The information is stored at a location that is preselected by the computer. The operating program guides the storage method, and it is up to the Macintosh—using the operating system’s guidelines—to always know where information is stored. All you have to do is give the information a name and it becomes a file, stored at a specified location on the diskette. Information is retrieved (read from) the diskette in a sort of reverse process: The read/write head searches for the named file by sorting through a list of storage locations (by file names) and then drops down close to the spinning diskette (it never actually touches the magnetic surface) to read the data.

The diskette recording method is not much different from that used in a tape recorder to record sounds. And, like the tape used in a tape recorder, a diskette can be recorded on, erased, and recorded over many, many times. The important difference is that while a tape recorder puts its magnetic message onto the surface of the tape streaming under the recording head, a diskette recording head selectively puts its recorded message anywhere on the diskette, as directed by the computer. We say that a tape recorder records (it actually reads the recorded message first and then reconverts it into recognizable sounds) serially; that is, it lays down its message in a long strip and therefore has to retrieve it in the same way. A diskette, by comparison, lays down its recorded message at any available spot on the spinning disk, at an address that is selected by the computer (the computer keeps track of available spaces and how much data they can hold).

The computer keeps track of everything by storing it all according to a storage address. The diskette is broken down into concentric circles, called tracks. Part of the first track is reserved for use by the operating system, but the rest is all available for files and file management records. When the computer gets ready to store or retrieve information, it only needs to tell the read/write head to move to one preselected position and hover over the surface of the spinning disk. Each of these tracks is divided into sectors. The division into sectors and tracks is actually done by a program in the Macintosh’s operating system. The program has to be run before you can use a diskette. When you first put a new diskette in the drive the program is called to format or initialize the diskette. To summarize, when the computer wants to put information on the diskette, it directs the diskette drive’s read/
Disk Drives and Diskettes

write head to a specific track, and then activates it exactly at the right moment—when a specific sector passes under the head.

Diskettes are said to be either single-sided or double-sided. The double-sided diskettes hold a lot more information than the single-sided ones, but you can’t just go out and buy the larger capacity diskettes unless your computer is made to use them. It depends on the design of the diskette drive. The initial model of the Macintosh came out with single-sided drives that used diskettes that held about 410,000 bytes of data (after formatting). Despite the fact that these diskettes are only single-sided, they actually can hold quite a bit more data than the older 5 ¼-inch double-sided diskettes could. How can this little single-sided diskette hold more than larger floppies? The answer lies in the design of the Mac’s diskette drives. The Macintosh diskette drives are designed to spin the diskette at speeds of between 400 and 600 revolutions per minute. The 5 ¼-inch diskette drives are designed to operate at a fixed speed of 300 RPM. The larger capacity of the Mac diskettes can be explained by this difference. To understand the way this works, you have to remember that the closer the head gets to the outer edge, the farther it is around one of the tracks. The Macintosh has a special way of changing the speed of the diskette drive motor as the read/write head moves farther out toward the edge of the diskette. Because of this, the Mac can store larger amounts of data on a smaller diskette, about 400,000 characters on a diskette as compared with the 360,000 that can be stored on the double-sided 5 ¼-inch diskettes.

If you look at most diskettes, you will see a square notch in the upper-right-hand area. This is called the write-protect notch. If you want to prevent the computer from writing anything on the diskette, you can cover this notch with a piece of tape (when you buy a box of diskettes it comes with a packet of little square tabs for just this purpose). The computer can still read from the diskette—it just can’t put any new information on it.

But the small diskettes used by the Mac have a different (and much niftier, we think) write-protect scheme. Hold up a Macintosh diskette with the metal part at the top and the arrow pointing up. Now look carefully at the lower right corner. See that little cut-out square area? The Macintosh diskette drives will test that area of the diskette to determine if it is write-protected. If the little tab is slid toward the bottom edge of the diskette, it is write-protected. Flip this tab into the write-protect setting on any diskettes that have important files that you want to be sure you don’t accidentally write over. But remember, if you write-protect a diskette, you won’t be able to store any new files on it until you eject the diskette and flip the little tab back up again.

Diskettes—even the tough diskettes used by the Macintosh—should be protected from extremes in temperature and kept away from strong magnetic fields. The jury is still out on x-rays, such as the kind used to inspect your hand-luggage in airports. We have intentionally let several diskettes pass through these checking stations, and they have come through with no damage. But many people think these machines can scramble the information on diskettes (if you show up at the hand-luggage checkthrough at Tokyo’s Narita Airport with a box of diskettes in your hand, they send a fellow running to take them and hand carry them around the machines; they don’t even take them through the metal detector). This may be unnecessarily cautious; however, we will admit that if a diskette contains important data, information for which we have no backup copies, we get a little nervous about those machines and prefer to hand carry them around the x-ray ma-
achines (the master copies of this book stored on diskette were so carried). Most airport security people seem to understand.

As we said, the main reason Apple chose the newer hard-cased diskette was because it is much more durable. If you look at a floppy diskette you will notice right away that you can actually see (or touch) the surface of the recording medium through holes in the plastic case. If you buy this type of diskette, you will be warned that you should never touch the surface of the recording material. You also have to keep it in a protective envelope as much as possible. The idea is to keep dust, oils, water vapor, and smoke from settling on the diskette's surface. Such microscopic things as dust particles may seem inconsequential to us, but to a read/write head, hovering barely off the surface of a diskette that is spinning at high speed, it might seem like a boulder on a race track.

The Macintosh diskettes don't need anywhere near this much protection. They don't even need any kind of protective envelope; the hard case is protection enough. You can drop them, sit on them, or even carry them around for days in your shirt pocket. There are only a few things you want to be careful not to do: Don't let your diskettes get too hot (leaving them in direct sunlight can distort the plastic), and don't let them get too cold either; the only other thing to watch out for is strong magnetic fields—they can erase or scramble the data stored on the diskettes.

BACK PANEL

Take a look at the back of the computer. Along the bottom you will see a row of sockets, each accompanied by a label and a pictorial representation of what plugs into that socket. The socket on the left is where the mouse plugs in. The next socket on the right is for an extra diskette drive. Then comes the printer outlet, and on the right is the outlet for the modem connector. There are a few other outlets on the back panel that are not used often, including one used to plug in a sound tape cassette. This can be used for the rare program that ties a sound recording to screen displays (Apple showed us a program that included an introduction to the Macintosh with an explanatory sound recording).

INSIDE THE BOX

That about covers it for the things that are on the outside of the computer. What about all those things lurking inside the box? All that electronic stuff, the wires and circuits and integrated chips? Although the design of computers electronics is a very complicated subject, the actual structure is not so mysterious.

THE SYSTEM BOARD

Let's take a brief look at what lies inside. If you pull the top of the Macintosh and look straight down at the floor of the computer, you will see a plastic board nearly covered with
wires and circuits and chips. This is called the *mother board* or the *system board*. The design of the Macintosh system board is somewhat unusual in that Apple chose to incorporate a variety of features onto this one board, rather than to create a set of replaceable boards for each operating feature. However, with the Macintosh, Apple tried to minimize the internal hardware components: to use less chips, lower voltages, and to come up with a design that avoided that big, noisy ventilating fan that has become so common on most personal computers. PCs usually make so much fan noise you can’t hear someone calling to you from the other room. The Macintosh operates in eerie silence.

We really don’t need to know much about the hardware that is part of the system board. There are a variety of chips that control the components of the system. There is the Motorola 68000 chip, the central processing chip, that does most of the data handling and the mathematical functions. There is an electronic clock. There is a chip that controls the serial communications ports and one that mediates between the main processing chip and the input devices such as the keyboard and the mouse. There are also the RAM chips that make up the Mac’s memory and some Programmable Array Logic chips (PALs) that keep track of the many signals that travel around inside the Macintosh. The motherboard handles most of the data input and output, including diskette access. It contains all of the memory and the controller for the communications ports. It handles the Mac’s sound generator, and the input from the keyboard and the mouse. It also has all the circuitry needed to control two diskette drives, even if you only have one.

**THE ANALOG BOARD**

Now, if you find the other board with circuitry on it—the one that is placed vertically in the box—you will be looking at the *analog board*. This board is where the power supply and the screen handlers are located. It also includes the necessary power switching devices and protection for variances in your wall-current power source. There is also a speaker.

If this list of what lies inside the box seems overly simple, it’s because the design of the Macintosh makes it that way. Missing are the usual set of controller boards, each with its own switches to set. In fact, there aren’t even any switches on the system board to reset for changes in memory capacity or for the communications ports. Apple is betting on this simpler design to cut down servicing problems and to make the machine more reliable in general. With many other personal computers, you have to take off the cover and actually get down inside the machine to set switches or to change the communications addressing. The Macintosh is not designed for that type of user. You can get inside there and prowl around if you want to, but for most users the inside of the box will always remain off-limits. For them, if the hardware is doing its job it should always remain invisible.
A CLOSER LOOK: THE OPERATING SYSTEM

The most interesting thing about the Macintosh's operating system is that most people don't know it has one. To most users, the way the Macintosh acts when you use it is just the way the Macintosh is. When users describe what is special about the Mac, they tell about the way it shows those little pictures (called icons) on the screen, and how it uses those menus that pull down from the top of the screen like you were pulling down a window shade. Or they will describe how those things, and others, make the machine easy to use. To them, those features are the Macintosh. Somehow, all that is easy to use about the Mac can be seen as being part of the computer itself; that is, part of the hardware. You go down to the store and buy a Mac. You open up the box, plug the thing in, insert a little diskette, and it runs. How it acts from then on soon becomes part of normal computer operation; it's just, well, it's the Macintosh.

The fact is, all those things are really part of the operating system, a computer program that theoretically could be run on a different type of computer. All of those screen messages and all of those little pictures are produced by a highly complex computer program that was designed especially for the Macintosh. This collection of complicated programs is called "The Finder." The Finder is the result of many years of programmer effort that must have cost Apple a great deal of money. The software development aspect of the Lisa and Macintosh projects took on a disproportionately large role. To develop this very special operating system, Apple threw away all of the existing rules and started from scratch. Apple didn't use one of its older operating systems and modify it. And, although
An Easy-To-Use System

Apple could have decided to go with a standard operating system, like the C/PM program, or the MS-DOS program that is a key element of the IBM PC standard, it decided to go its own way. Had Apple chosen the usual direction there would have been essentially no software development costs: Apple could just have bought the program from somebody else who had already done the development.

But Apple decided to do it right. It decided that those other programs were not good enough. Apple’s basic philosophy was that almost all of the other operating systems required advance learning: Before you began to use the computer you had to read a book about the operating system. Apple wanted a system that would be so logical, so self-evident that almost no reading would be required. That’s why Apple chose to use the mouse, and began to develop the pull-down menus, things like the built-in calculator program and the control panel. That philosophy, based on concepts developed at Xerox’s Palo Alto Research Center in the 1970s, led to designing an easy-to-use system, and that led to other innovations, like the status boxes that tell you what you have done so far and ask you what you want to do next.

The result of all Apple’s effort is the Finder, a collection of programs that are so well integrated within the Macintosh, that they appear to be part of the hardware itself. Although we may accept the idea that it makes sense for other computers to use a purchased operating system, a standard system that is very similar to the operating system on other computers, we wouldn’t accept that approach for the Macintosh. To anyone who has spent some time with the Mac, the operating system has become a reliable friend; in many ways, it is the Macintosh.

AN EASY-TO-USE SYSTEM

When people speak of why the Macintosh is different from other computers that came before it, they usually refer to how easy it is to use it. They say it is easier to use because you can use the mouse to point to what you want to do next. Each step is selected that way and every new program is presented the same way. You never have to learn a new method of choosing the program’s steps. The reason the Macintosh works like that is because the operating system provides a uniform working environment. The Finder program shows you the same screen every time you load a new program; the icons tell you the main program choices (things like which files are available) and the pull-down menus are always there for selecting program details. The point is, once you learn the basic system, every new program can be run using the same general methods of operation.

Also, the Macintosh is less confusing to use because it lets you choose what you want to do by picking choices from a group of pictures or from a simple list instead of making you memorize strange commands that have to be entered in exact sequences.

Each of these things contributes to making the Macintosh what it is, but none of these things are attributes of the computer itself; they are all features of the operating program. The truth is, even though the Macintosh’s hardware is well matched to this type of operating system, such operating programs could be modified to run on other computers. Computers such as the Apple Lisa and the Xerox Star use operating systems very
much like the Macintosh's Finder program. And there are other operating systems similar to the Finder that can be used on a great variety of different computers. They emerged not long after the Macintosh when the brisk early sales of the Mac proved that this type of operating program would be accepted. Some of them even use a mouse and icons on the screen.

So what is it that makes the Macintosh unique? Why do so many see it as a "revolution" in computer design? We think it is because the computer and the operating software were designed from the ground up to provide functional but friendly user access. By that we mean that when the original choices were made, when it came time to decide where the tradeoffs would go, the design team chose to make everything easy for the inexperienced user. So what if pictorial error messages took up a lot more overhead (more memory and computer operating time)? If it helped to make the machine easier to use, the designers decided to go ahead and put in the pictures. And if keeping those pull-down menus available at all times took up a lot of extra screen memory, so be it; it kept the user from having to remember how to call up the menus by command.

If you are designing a computer, as soon as you begin to make those kinds of choices you start a chain reaction of other design choices. Clear pictures on the screen require higher resolution graphics. This means a different type of cathode-ray tube (CRT) and a more complex design for the graphics controller. If you don't want the user to have to be involved in saving programs to the diskette (other than saying, yes, I want to save), you have to design a much more complex operating system that uses up more memory and takes longer to execute. Those are basic design choices. And Apple had faith in its overall design philosophy. It wanted to make the Mac simple to use and it was willing to risk essentially the whole company by spending the time and money to make the Mac exceptionally simple to use. The size of the computer, the simplicity of the keyboard and the mouse, and the methods of diskette loading and ejecting all broke with the "standard" concept of what a computer should look and act like.

And it turned out that Apple was right. People did want a computer that was easier to use, a more powerful computer that gave the user a lot of powerful software that could be used under one consistent operating environment.

The key to the Macintosh's success is in its operator interface, the point at which the user interacts with the computer.

THE USER INTERFACE

Let's look more closely at that idea. What is a computer user's view of the computer? When we press keys on the keyboard do we think of the scan codes passed to the computer's processing system? Do we consider how that signal is acted upon by the computer's electronics? How it organizes the input and activates the electron guns to create a representation of that key's decoded value on the screen? No, we just press the key and expect the corresponding letter to appear on the screen. We computer users see our task as merely to select the proper key to push it. In response, we expect the computer to do its job.

The point of operator interface is where the user perceives it to be, at the keyboard,
at the mouse and the display screen—not somewhere deep inside the computer. The people at Apple were aware of this when they designed the computer. That’s why they designed a simple keyboard. That’s why they adopted the mouse, and that’s why there is only one button on it. You can use the mouse to point at what you want and then press the button to select it. This method of interacting with the computer creates a new and simpler user interface. Apple saw the interface that involved typing in a set of commands and special symbols was too complicated. Users were being required to memorize complicated command structures that changed with every new program. Why not provide one simple set of interface operations and one simple set of easily selected commands that don’t change as programs change? It’s that basic strategy that led to the development of the Macintosh operating system.

THE NATURE OF OPERATING SYSTEMS

To get a better idea of what the Macintosh’s operating system has in common (and what it doesn’t) with most computer operating systems, let’s look at how most computers handle the user interface.

First, we should be clear about what operating systems do. If you turn on a computer without having some sort of operating system, the computer will just sit there. All you will see on the screen is some sort of error message. Most computers have a tiny bit of an operating system built in; usually just enough to give it the ability to read information from a diskette. The idea is that the real operating system information will be on a diskette and you, the user, will know enough about how computers work to put a diskette into the diskette drive slot. Most computers have a small built-in screen message producing ability so if the computer doesn’t find a diskette in the drive it can present some kind of message to you.

The Macintosh does this too, but its built-in kernel of a program is more complicated. Its program displays a picture of a diskette on the screen with a question mark on it. Other computers, presented with the same situation, only display an error message. If they don’t find a diskette in the drive, they display something like:

FDD ERROR 214-A

That kind of message is enough to convince most computer newcomers to never turn on a computer again. That kind of message basically says: ‘‘Hey, dummy, you have done something wrong, and you don’t even know what it is.’’ And unless you know what a 214-A type of error is, you will probably just turn the computer off and sneak away hoping you won’t be billed for any damage. The reason most computers act like that is because they were designed by people who do know what a 214-A type of error is. It may never have occurred to them to think that someone else might not know. That’s because for them the interface with the computer is somewhere down deeper inside the machine. The people who build computers know that when they press keys on the keyboard the computer acts via a system that scans the keycodes. They have the knowledge, the back-
ground, to allow them to think that way. For them, the most efficient way to tell a user (especially if they think of users as people like themselves) that they have forgotten to put an operating diskette in the drive is by displaying a 214-A error message. Such a message requires very little computer overhead; that is, only a few bytes of memory are required to store the message, and it needs only a simple fetch routine to bring it up to the screen and display it. The very idea of using all the computer overhead required to present a picture of a diskette with a question mark on it would seem absurd to them. Besides, if you look up the 214-A error in a table of error codes, it will give the knowledgeable user a lot more information than a picture can.

How did this idea of low overhead computer usage become so important? It's because the people who are designing operating systems today are the same people who designed the systems back in the days of slow processors and limited computer resources. The more tightly you wrote program code, the more efficiently you used the computer's processing time and memory, the better the whole system worked. In fact, a lot of today's operating systems are merely adaptations of programs that were written for those older systems.

But today, newer processing systems and faster hardware designs with much larger memories have made that kind of super-tight programming obsolete. High overhead routines hardly slow down the computer at all. The early personal computer operating systems were written for computers of limited processing power. They could only handle data either 4 or 8 bits at a time. And they had maybe 4K to 8K bytes of memory available. Compare that to the Macintosh's powerful 68000 processor, 32-bit data handling, and 128,000 bytes of RAM and you can start to see why the old operating system designs are no longer applicable.

**EARLY OPERATING SYSTEMS**

In the beginning, there were no operating systems. The computer users (of small computers, at least) were the people who had built them. They didn't need easy-to-use programs to gain access to the computer; they knew exactly how to talk to the machine itself. In the beginning, the user interface was right down at the bit-by-bit level. There was no screen and no keyboard; users told the computer what to do by changing switch settings. The only feedback was via little twinkling lights across the front of the machine.

It didn't take long for users to want more information about what was going on inside the box and to have a more useful interaction with the machine. So keyboards (based on typewriters) and CRT monitors (based on television) appeared. That made computers more accessible to less skillful users; the people using computers were no longer necessarily the ones who had built them. These early users were still very knowledgeable, or else they grew knowledgeable very quickly. Despite the improvements, computers were still not very easy to use.

These early users were constantly looking for ways to make their computers work better. For example, they noticed that certain computer operations were done over and over. Why not create a few programs to do these repetitive tasks and then when the task
needed to be done just call up that program? That way, they reasoned, they could go on to
discovering other interesting things without having to do the same repetitive things over
and over again. They wrote programs to read information from paper tapes or from re¬
cording tapes (later from diskettes). They began to discover the best ways to lay down
files on that kind of storage and how to retrieve them again.

And always there were the "outsiders." The outsiders were those pesky people who
didn't have degrees in computer engineering, but who despite that still wanted to use a
computer. These outsiders didn't know how to write an operating program and didn't
want to learn how, but still saw some ways that computers could make their tasks easier.
These users heard about nifty programs the insiders had already written. When they
could, they talked the experts into modifying these programs to run on their computers.
Thus the first rudimentary operating systems were born. When enough people were bor¬
rowing these programs, the program writers started to make a few bucks by selling them.
The programs were modified to run on a variety of machines, and as sales picked up the
capabilities of the programs were expanded. Before you knew it there existed a number of
standard operating programs that gradually had become complete operating systems.
These early operating systems were carefully tailored to individual machines. Later, as a
few of these early computer types (like Steven Jobs and Steve Wozniak, who built the first
Apple computer in their garage) began to sell small computers to hobbyists, they provided
such operating systems along with their computers. These systems, specifically designed
for one computer, were either written by the computer designers or by programmers that
the designers hired. That way, the programs belonged to the computer company and were
copyrighted to keep any other company from using them. But other manufacturers could
hardly have used these proprietary systems anyhow: they were so closely tailored to the
individual computer, and each company was making computers so different from the oth¬
ers, that the operating systems would have needed to be entirely rewritten.

But few of the early microcomputer makers were much interested in making appli¬
cations software for their machines; they were in the computer hardware business. Pity the
poor users who discovered exactly the application they had been looking for, only to learn
that it was designed for a different computer. What such users may not have known was
that the only thing that kept them from using these applications programs on their com¬
puter was that the program had been written to operate under a very specific type of
operating system. One thing soon became clear: The way to make a lot more software
available was to make more portable operating systems. If someone could design an
operating system that could be run on several computers, the same programs could be run
on all of them.

One of the first portable operating systems was designed by Digital Research, a
small software company with a good idea at the right time. Digital named its program
CP/M, the Control Program for Microprocessors. This made it easier for manufacturers.
All they had to do was purchase the rights to CP/M and make the minor modifications to
adapt the operating system to their computers. The only problem was that the program
was designed specifically to run on computers that used the 8-bit Intel or Zilog
microprocessors. Some companies like Radio Shack and Apple chose to stick with their
own hardware designs and operating systems. And for a while those were the main
choices in the microcomputer world: gain access to more software by choosing a computer that could use the CP/M operating system, or go with one of the bigger companies like Apple (using the AppleDOS or ProDOS operating systems) or Radio Shack (using the TRSDOS operating system) that also came with their own catalog of software.

But something was going on in the world of small computers: people were outgrowing their limited-capability 8-bit computers. The move was underway to the faster 16-bit machines. Digital research soon came out with a 16-bit version of the CP/M program but there was a new player in the game, the giant IBM company with its new IBM PC. IBM chose not to create its own operating system, but instead of choosing the CP/M program that appeared to be the emerging standard, IBM worked with the Microsoft Corporation to develop a new operating system (very much like the CP/M program), called PC-DOS (PC Disk Operating System). Like the CP/M program, this operating system came on a diskette and was loaded into the computer when it was first turned on. Microsoft began to sell a version of the IBM operating program to other computer makers and for a while there was a seesaw battle for customers between Digital Research's CP/M and Microsoft's MS-DOS. But with the great success of the IBM PC, other hardware manufacturers began to emulate the IBM and MS-DOS became something of an industrywide standard. Later, IBM also began to provide a version of CP/M for the PC, making the PC one of the first machines to give users a choice of operating systems.

THE NEW GENERATION OS

As logical as these portable operating systems are, something new is now beginning to happen in the small computer world. As small computers grow more powerful, the portable operating systems are straining to keep pace. More powerful processing systems and much larger memories are harder to control with the aging DOS systems that were originally written for the old 8-bit computers. Because of this, a new generation of operating systems is emerging. These systems all have one thing in common: like the Finder program used on the Macintosh, these programs provide a simpler, more understandable interface between the operating system and the computer user.

If the power of the more complex systems is to be made available to users, a simpler way of interacting with the computer has to be found. Since users can no longer be expected to memorize all the complex commands for all the functions of the more complicated operating systems, a shell is needed to overlay the actual functions. This shell can serve as a buffer between the user and the complexities of the operating program. It should be thought of as a user's package that surrounds the functions of the operating system. By using the shell, users can still have access to all the system's capabilities, but with a more understandable way to interact with it. The shell both gives the user access to the capabilities of the computer and protects the user from the complexities related to that greater capability. Menus are one example of a good shell technique. They let users choose the desired operating system function by letting them choose what they want to do from a list, or menu. The menu does not necessarily describe the capability of the operating system, but instead describes some of the things the user might want to do. It is
up to the shell system to translate the user's wishes into a coded set of commands for the operating system.

For example, let's say you are using the computer and you want to print out a copy of what you are working on. With a command oriented system, you would have to know how to use the operating system's print command, along with a complex set of rules about how to enter the command. You would have to know how to stop what you are doing and get into a different mode, one in which you could send commands to the printer without losing the file you were working on. Then you would have to know the correct syntax to enter in order to get a certain number of copies printed, to determine which pages of the file to print, or to specify a large number of other choices regarding margins, paper length, etc. Using a shell system, you only have to select the print function from a menu and then select your printing choices from another menu. The shell has all the built-in codes to tell the computer how to print the file; you only have to tell it what you want to do.

In the last few years, such shell-type approaches to operating systems have been quietly under development. One of the earliest and most innovative was developed at the Xerox Corporation's Palo Alto Research Center (PARC). In the 1970s, Xerox had assigned teams of researchers to investigate new ways to approach the operator interface with small computers. These researchers chose to move away radically from the traditional, command-oriented operating systems and concentrate on methods designed to "protect" the computer user from the complexities of system operation. The first commercial result of this research was the early Alto computer and then the more marketable Xerox Star. Because the Star was (and is) an expensive, highly sophisticated computer, few ever saw one. It, rather than the Apple Lisa and Macintosh, was the first computer to use a mouse and on-screen icons and menus.

APPLE'S NEW OS

When the operating system developers at Apple saw early versions of the Xerox machines, they knew they were seeing the future. About this time, Apple went to work on its version of the mouse/icon/menu operating shell. It took 200 programmer man-years (development time equal to one programmer working for 200 years or 200 programmers working for one year) to develop the Finder program. It also took a highly sophisticated hardware design to run it because the overhead cost (in terms of computer resources) was so high that a really protective shell would slow the computer's operations down so much the program's usefulness wouldn't be worth the added wait time.

Strictly speaking, the Finder is not really an operating system. Think of it as the shell that covers the real operating system. An arguable difference perhaps, but important if we are to understand what is new about the new generation of operating programs. In the printing example we described above, it is the role of the operating system to talk to the printer, to pass along all the correct codes regarding printer function. It has to know where to find the file, how to convert it to the proper format in order to communicate with the printer, how to tell the printer about print quantities, margins, paper lengths, and so on. With some earlier operating systems, the user had to know enough about the system to
be able to specify all those choices. But there is no real need for the computer user to know all those things. Why not just let the user point to the choices on a menu? This is the role of the shell: to organize the information in an understandable way and make the user's choices easy.

There were some other considerations that Apple had to face when deciding to redesign its operating system. Apple had to make a tough choice: design an entirely new operating system that is easy to use and that takes advantage of the modern computer designs, or go with an existing operating system and gain the advantages of a familiar program with existing software. Sounds like an easy choice? The thing that makes it not so easy is the desirability of supporting existing software. As we pointed out in our chapter on buying a computer, the one thing that ultimately sells a computer is the programs that are available for it. At the time Apple was starting to design its new computer, the rage was IBM-compatible software. With the IBM PC standard there was finally one type of operating system that was becoming known to all computer users; there was a tremendous amount of software made for it. Almost anything you wanted to do with a computer could be done with one of the existing IBM PC compatible programs. Should Apple go against the conventional wisdom of the times and give up all those programs that might help sell the new Apple computer? Most of the other computer companies were choosing not to fight the IBM tide. Even Radio Shack, Apple's traditional foe in the low-priced computer arena, came out with an IBM-compatible machine. Many expected the rumored new Apple computer to do the same.

And Apple had another problem. Even though its overall share of the small computer market had been gradually slipping since IBM introduced the PC, Apple had held its own almost entirely through a base of customers faithful to the Apple II and III series computers, especially in the schools. Should Apple risk sacrificing the customer base by coming out with a new operating system that not one of all those thousands of programs for the older Apples would work with? But Apple had faith in its dream: it chose to ignore the IBM standard and even to sacrifice compatibility with the older Apples. Apple felt it needed to design an entirely new operating system with an entirely new user interface. The company was willing to bet all its resources on one computer that would be so useful that it would lure customers away from both the IBM standard and from even the existing Apple standards. It turned out Apple was right.

Perhaps there was even a greater risk in Apple's choice of a new operating system, one that most people might have overlooked. As we pointed out earlier, all computer companies, to some degree, live or die based on their acceptance by the third party software producers. No computer hardware company can create enough computer programs to keep all their users happy. There are just too many specialized needs. So a lot of the success of a new computer depends on whether or not the software producers get on the bandwagon and begin to support it by creating new programs for it: the more new programs that are available for the computer (and the faster they become available) the more people are likely to buy the machine. In the case of the Macintosh, the key was whether or not the software producers would be willing to support the Mac's new operating system.

The problem was that software makers had never seen anything like it. When Apple
first introduced the basic system on the expensive Lisa computers, very few software makers were willing to support it with new programs. Not only did the computer sell poorly, but its new type of operating system was extremely complex and demanding. With it, Apple seemed to be making a strong statement: All software was to have Apple's ease-of-use shell. This made producing programs that would run under the Apple operating environment all the more difficult. Even when Apple came out with the less expensive and more marketable Macintosh, the software producers were still faced with a tough decision: If they were going to make programs for the Macintosh, they had to redesign their existing programs. Most application of software on the market had been written for either the home computer market (often designed for simple 8-bit systems) or for the business market (usually for IBM compatible, 16-bit systems). To rewrite these programs for the 16/32-bit Macintosh, with its much more complicated user interface, would constitute a major programming effort. And major rewriting efforts like that take time. In the commodity software market, the first ones out with a new application often capture the market. It’s very tough for latecomers to break in. The longer the Macintosh was on the market, the more precarious was the position of the software developers. Apple, in forcing a new operating system on the software developers, was taking a big risk. If the developers refused to put in the effort, the Macintosh might fail. Apple could be stuck with a cute little machine sporting a very effective operating system with no software to run on it.

But this seems to be a case of all’s well that ends well. The Macintosh caught the public’s fancy because of its ease-of-use features (in other words, the easy-to-use operating system) and the software makers quickly jumped in to support it. There was a lag time of several months before anyone was sure how much software would be available, but pretty soon it started to pour in. And since the buying public seemed to like the user interface represented by the Mac’s operating system, the software makers chose to give their programs the same type of user interface framework. This made the larger set of Macintosh programs appear more unified.

THE DESKTOP CONCEPT

When Apple decided to utilize one of the new generation shell-type operating systems, it looked for an overall concept that would make it easy for users to interact with the system. What should users see on the computer screen when they use the Macintosh? Apple decided they should see something that looked like a normal desktop. Does the Macintosh’s screen really look like the top of a desk? Not really. In this case, desktop refers to a concept: a way of working with a computer that matches the way a person would work at a well-organized desk. The screen becomes a working area onto which you, the user, can place any or all of your working tasks. If you want to open up a text file (say you want to write a memo to the boss), you can use the mouse to click open a new file folder, name it BOSSMEMO and start working with MacWrite. But then, while you are writing the memo, you decide you want to include the latest cost figures so the boss will see how
much money you have saved for the company. Do you save the BOSSMEMO file, put it away, and open up a new file in order to use the computer as a calculator? Not if you are using the Macintosh. You just pull down the menu and select CALCULATOR. The calculator appears right on the screen. You can use the mouse to press the calculator keys, or you can enter numbers and arithmetic operations right from the keyboard. You can add up all the numbers, calculate the total, and then type the results into your memo to the boss (it's still on the screen). When you are done with the calculator, you can put it away or, if you want to use it again soon, you can just drag it over to the edge of the screen and leave it there.

If you are familiar with the use of the Macintosh, this is probably just an ordinary computer task sequence. But what you may not know (unless you have used other computers) is that you have just accomplished something that should be impossible. You have asked the computer to do two different types of tasks at the same time. The computer was still using part of its resources to keep track of the boss's memo (remember, you hadn't saved it to the diskette yet) while you asked it to go call up a different type of program (the calculator program) and do some mathematical calculations.

So, couldn't you have done this if you had been using an ordinary computer? Macintosh users will grasp at how troublesome this apparently simple operation can be, but if you were using another type of computer you would have had to stop working on the memo to the boss, enter a special save command, and wait while the computer saved the file on the diskette. Then you would have had to call up another program, or maybe even take the diskette out and put in a new one with a program that is designed to do mathematical calculations. When you were done calculating you probably would have to write the result down on a piece of paper (what a thought, writing with paper and pencil when you have a computer sitting in front of you!), put away the math program, reload the word processing program, call up the file called BOSSFILE, and enter the numbers that you (hopefully) can read from the piece of paper (if you haven't lost it or written over it when the phone rang).

The reason the Macintosh allows you to do two (or more) tasks at the same time is because the computer is designed to do multitasking. That is, the computer has the resources to keep track of one or more files while it acts on other tasks. That is a very special advantage of the Macintosh. It is due both to the advanced design of the computer and to the powerful processing system that combines an expanded capability processor with Apple's innovative desktop concept design. It allows you to use a computer in the way most people normally work at their desk; that is, frequently moving from one task to another. At your desk, you can be working on a task and then decide you need to work on another, as in our example of working on a memo and discovering the need to do some mathematical calculations. That's the way most people work, and the designers of the Macintosh took that into account when they began to design the computer.

Since the desktop concept was introduced with the Macintosh, many people think it will become the standard. It provides an easily understood method of interacting with the computer lacking in most other computers. To use the desktop concept, you take actions for which you already have a model. The way you use the computer is not so different from the way you do other tasks. You retrieve things by pointing at their picture, you get
The Desktop Concept 51

rid of them by dragging them over and dropping them into the wastebasket. The Macin­
tosh uses concepts you are already familiar with and so makes the computer interaction
conform to your working methods (instead of the other way around). It has been said that
the most difficult thing about learning how to use a computer is to learn “how the com­
puter thinks.” That used to be a very important step in learning to use a computer. If you
didn’t understand how the operating system worked, you could get very confused by the
system error messages, the very things that were supposed to help you straighten out your
problems. The problems that came about while using a computer were almost always
caused by the user doing something wrong. But the hard part about using most personal
computers was that the error messages reported trouble in terms of operating system func­
tions: you had to understand how the operating system worked in order to understand what
the error messages meant! You almost had to know what you did wrong in order to correct
the problem. But why would you have done it wrong if you knew it was wrong and knew
how to do it right?

The Macintosh, on the other hand, assumes you don’t know how the operating sys­
tem works. Or at least it assumes that it doesn’t matter. The idea is that you shouldn’t
have to know much about computers in order to use them. The computer should be smart
enough to know how you work and make it easy for you. For this reason, the operating
system presents you with a limited set of choices, usually yes or no choices. Do you want
to save this file? Yes or no. If yes, do you want to assign a name? Yes or no. Sometimes
the Macintosh just explains what has happened and your only response is to move the
pointer to the OK box and click: as if to say, OK, I understand.

Some experienced computer users may not like the shell that surrounds the Macin­
tosh’s operating system. The designers may have thought of it as leading the user along by
the hand, but some users will see it as being led along by the nose. They want more op­
tions besides yes or no. They want to be able to enter a large number of different com­
mands and thereby have the option of carrying out a variety of operations at any time.
Actually, even though most computers can’t act like the Macintosh, the Macintosh can act
like other computers. The operating system gives you the option of using some of the
Mac’s special keys to take a shortcut to many of its capabilities. The key with the four­
leaf-clover symbol ( ) and the OPTION key can be combined with other keys to
carry out many functions.

Still, the Macintosh can’t mimic the command-oriented nature of most computer
operating systems. With those systems, you enter a command to begin a process. Then
you often enter more commands to continue. Each command has an accompanying capa­
bility and a related bit of programming to carry out the task. The Macintosh is really no
different in that regard: each of the entries on the pull-down menus is like a command in
other computer operating systems. It’s just that the menus give you easier access to the
programs, and the advanced processor inside the Macintosh acts on the menu choice very
quickly. So quickly, in fact, that it’s hard to realize that the operating system has had time
to go off and fetch some new programming code. When you ask the computer to display
the alarm clock (by selecting ALARM CLOCK from the pull-down menu), the clock ap­
ppears so quickly it’s hard to realize that the operating system had to load a special software
program and act on it before the clock could be displayed.
And the desktop concept seems to be catching on. Because of the success of the Macintosh, other programs are now being written that use the desktop concept as the organizing principle of the user’s interface with the operating system. They also use the “windows” concept, which allows you to gain access to other of the computer’s resources while working in a file. The windowing idea is that the screen is normally a window into the computer’s operation: If you are using a word processor to type out a book, the screen is a window that shows a limited part of that book, about one page full of text. But what if you want to use another of the computer’s programs? Then the screen can be split up into more than one window. You could create a smaller window that will still show you part of the text you are working on, and another window that will give you access to a different program at the same time. Using the Macintosh, the calculator program is an example of a second program brought onto the screen at the same time as a text writing program. If you think of the calculator display as a window that lets you look into the calculator program while you are working on the memo writing program, you can better understand the windowing idea. This may seem simple, but it actually takes a great deal of complicated programming to make it work, and it takes one of the fastest, most sophisticated hardware designs to make it all work fast enough to be useful.

There are two ways to carry out this multitasking programming. One way is to simply make more than one program available in more than one window on the screen. With this type of operating system, you can use two or more programs alternately. In other words, you can jump back and forth between programs without having to go to all the trouble of saving the program and then retrieving it again. But what if you want to start a complicated program running, and while it continues to run go off and work on another program? This requires a more complicated operating system. It has to keep track of what is going on in the background while giving you access to the files or programs you are working on in the foreground. This requires not only a more complex operating system, but also a processing system capable of carrying out more than one task at a time. This required mix of multitasking hardware and software is what is represented by the Macintosh, and it is this very complexity that makes using the computer much less complicated for the user.

FUNCTIONS

It isn’t the operating program Apple calls the Finder that is so different; it is how you, the user, relate to it. All operating systems do about the same things. That is because there are a certain number of things that have to be done on all computers to keep things in order. For example, a computer operating system (including the Finder) has to:

- Format diskettes by mapping out the available area on them in a way that helps the system keep track of stored data.
- Keep track of and display what is stored on the diskette.
- Name, rename, and erase files.
- Copy data from stored files to other parts of the diskette or to other diskettes.
- Load files from the diskette into the computer’s active memory.
There are a number of other things an operating system needs to do, but these are the essentials. And the Macintosh is no different: Even with a computer as easy to use as the Mac, you still need to prepare diskettes and manipulate files. But with the Macintosh, the operating system makes it easy.

By the way, there isn’t only one operating diskette; it’s just that there is only one set of operating programs. You can transfer this set of programs to any diskette you want. You see, Apple wrote the program that guides the basic operation of the computer, and it is willing to let other software developers put that program on many different types of diskette-based programs. Some of the programs that work on the Macintosh (even ones made by people other than Apple) have that same operating program—the Finder—on the diskette along with the applications program. That’s why the Mac acts the same as usual when you insert the diskette with the new program into the drive.

And what is acting “the same as usual”? Let’s review what happens when the operating system is loaded into the computer’s main memory:

**THE DESKTOP**

When the operating system is loaded, the screen changes to look something like this:

![Desktop Screen Image](image)

This screen is supposed to look like the top of an ordinary desk (it will look more like a desk when it gets all cluttered up with files, calculator, a clock, etc).

**PROGRAMS AND WINDOWS**

All of the available programs for the Macintosh are represented by some visual symbol displayed on the desktop screen. But then what happens to this desktop screen when one of the programs is activated? Answer: that’s up to you, the user. One of the innovative things about the Macintosh is that each of the available programs can occupy space on the
screen. As we said, while the program is inactive it is represented by a visual symbol on the desktop. When most computer programs are activated, they usually take over the whole screen. And this can be true with the Macintosh too; but, with the Macintosh, there can often be situations wherein more than one program can occupy the screen at one time. When a program is activated (refer to the descriptions of screen icons and clicking below), one or more windows will appear on the screen. These windows often hold information about available subprograms that can be activated individually. Another thing about windows: When a program has been fully activated it can still be relegated to window status on the screen, its size can be reduced, and it can be pushed to the side of the screen (or even clear off the screen) while another program is called into action. The power of the Macintosh allows two or more programs to be operational and on the screen at the same time. Each is active in its own window.

This is what a fully activated window looks like on the Macintosh screen.

![Active Window](image)

Compare that to an inactive window.

Notice that the differences between an inactive window and an active one are subtle, but important. The bars along the top of the window are highlighted and some arrows appear along the sides of the window and at the corners. Each of these features has a special use once a window is activated.

**Title Bars**

The bars along the top of the window are called title bars. They are often used to "grab onto" when you want to move the window to a new position on the screen. Refer to the section on "dragging" for more information on this feature.
Close Box

At the left side of this title bar area there is a small white square, the close box. The close box is present whenever a window is active. The mouse can be used to get rid of the window. Just position the mouse pointer over this box and press the button.

Scroll Arrows

The arrows that sometimes appear at the bottom or sides of a window are known as scroll arrows and can be used to "move" the window to reveal more. Think of the window as a hole that looks into the program; the window is not big enough to see very much of the program, but the scroll arrows allow you a way to move the window and therefore see more of the program's contents.

Size Box

You can even change the size of the window: just use the size box that always appears in the lower right corner of an active window. Position the mouse pointer over the size box and press the mouse button. Then, while holding down the button, move the mouse right and down to make the window bigger, left and up to make it smaller.

Pull-down Menus

Many other features of the windowing system are hidden in the pull-down menus. A menu is a list of choices currently available and there can be one, two or more of these menus
hidden from view. Each hidden menu will be designated by a label along the top of the screen, such as FILE, or GOODIES, or the one just labeled with the picture of an apple. To see what is on these menus, position the pointer over the label and press the button: The menu will magically appear. The menu will stay on the screen as long as you hold the button down. Then you pull down (slide the mouse back toward you) to select which item you want to select from the menu. As you pull down, the choices will be highlighted by a black background. When the choice you want is highlighted, let up on the button and it will be automatically selected.

**Program Menus**

When you load any program into the Macintosh’s active memory, there are a number of functions available that are related to that program. In addition to the pull-down menus, these functions may appear in another type of menu found at the side of the screen or along the bottom. Any of these functions can be accessed by using the mouse to point at the one you want. To access these, just position the pointer to a label (often a pictorial label—as in the MacPaint program), and press the mouse button. The menu may not change, but the function will be instantly activated. In MacPrint there are paint functions along the left side of the screen; you just move the pointer to the function you want and press the button. There are also painting patterns along the bottom of the screen. You select them in the same way, by just pointing, using the mouse, and pressing the button.

**Icons**

Icons are those little pictures that represent the files that are currently available. These files may be stored on the diskette or they may be already available in active memory. There are only a few ways to interact with the icons, and once you get used to them these methods become second nature.

**Pointing**

One way to interact with the icons (and with other screen features) is by pointing. By that we mean moving (also known as “rolling,” since there is a ball in the underside of the mouse) the mouse by sliding it along the top of your table or desk. As you move the mouse the pointer on the screen will move in the same relative direction. With practice, you soon will get used to the motion of the mouse pointer, and you will be able to quickly move it to exactly the right spot on the screen. It will become second nature: You just watch the screen while your hand zips the pointer across the screen.

**Clicking and Double Clicking**

You can use the mouse pointer to click on icons or other screen features. You just position the pointer over the item, press the button and let it go. That “selects” the file or function represented on the screen. By selecting it you have made that program the active program, the one that will go into action next if you tell the computer to load the program.
You can start that program in operation by pointing to a start message in a menu, or you can do it more quickly by "double clicking" on a screen feature. This means that you position the pointer over the item and quickly press the button twice. This both selects the program and starts it in action. Often starting the program in action does nothing more than show you a new set of icons, each of which represents a subset of the total program. Now you go through the same process again: Move the mouse pointer to the icon that represents the part of the program you want to run and click or double click to get it started.

**Dragging**

There is another way to use the mouse. You can drag screen features such as icons. Dragging is a mouse function that is done by positioning the mouse pointer over an icon (or other onscreen feature), pressing the button down and holding it there. Next, you move the mouse across the tabletop—still holding the button down—to a new position on the screen. Sometimes, you can use this dragging feature to combine the functions represented by the icon with other features represented by other icons. For example, you can drag an icon representing a file folder over to the icon representing a trash can. When you release the mouse button, that file will be “deposited” in the waste basket; in other words, you have just used the dragging feature to throw away a document you no longer need.

**Dragging Windows**

You can use the dragging feature to move almost anything that appears on the Macintosh screen. You can even move windows. A window appears when you activate a program. The window, then, is the onscreen space that the program’s display appears in. For example, if you select the MacWrite program by double clicking the MacWrite icon, a new window will appear on the screen with icons that represent the features of the MacWrite program. You can move this window (or any other) to a new position on the screen by moving the pointer to one of the edges of the window, pressing the mouse button, and holding it down as you roll the mouse across the table. A shimmering square that represents the window will move along with the pointer. When you get the square where you want it on the screen, release the mouse button and the window will instantly move to that new position.

**TRASH FEATURE**

One special feature of the operating system worth noting is the Trash program. This program is represented by a trash can, usually placed down in the lower right corner of the screen (but it can be moved to any spot on the screen—you decide). Macintosh users tend to forget that this trash feature is a computer program, as is everything else that appears on the Macintosh screen. The trash program is used to get rid of programs you no longer want or need. On other programs, this might be called the ERASE or DELETE program.
Like most other Mac programs, this program is run by moving the visual representation of the program. You just drag the program icon to the trash can and leave it there. For example, say you are using the MacWrite word processing program and you have rewritten a letter to your friend Linda. Let’s say you had labeled the original letter LINDALET and stored it as a file under that name on the diskette. But then you changed your mind and rewrote the letter, renaming it LINDALET2. You can now get rid of the original by dragging the icon labeled LINDALET to the trash can icon. But even though you have dropped it into the trash and the LINDALET icon no longer appears on the screen, it is still stored somewhere on the disk. To get rid of it for good, you have to select the EMPTY TRASH item from the “Special” menu. But be careful what you drop into the trash: important files like the MacWrite program itself can be erased if you drag them to the trash and then empty the trash later.

If you make a mistake and put something in the trash can that you didn’t mean to, just double click the trash can icon and then—when the trash contents window appears—double click the item you accidentally threw away. It will be restored to its proper place on the disk and its icon will reappear on the desktop screen.

BUILT-IN PROGRAMS

Most of the programs built into the operating software are almost transparent to the user. You can do a great many tasks without really realizing that it is the operating system that is controlling the action. For example, there is no real need to understand that the operating system includes a special program that copies files. Or that there is a built-in program to save what you have done in a named file on the diskette. These programs do their job with very little input from the user: Usually, you just say go do it (by pointing and clicking with the mouse), and before you know it, the job is done.

But there are some special programs—called desk accessories—that are built into the Macintosh operating system. These may be used more often than the file maintenance programs, but, because they are designed to mimic some actual desk accessories, many users may not think of them as operating system utility programs. The desk accessories are not largescale applications programs, just programs that Apple has included with the operating system that can help you do your work. Some of them are based on the kinds of things that might be found on a normal desk, and others have more to do with maintenance of the computer system. They are so interesting in their own right that we think it is worth getting to know how to use them. They are described below.

Calculator

This program, loaded from the pull-down menu at the top of the screen (labeled with the little apple) displays a “window” that looks just like a handheld calculator. When it first appears, it is more or less in the middle of the screen. But you can drag the whole thing off to the side of the screen if you want to keep using it while you are working on something else. You drag it by positioning the pointer over the black bar along the top of the calculator, and then holding down the button while you slide the mouse across the table top. The
calculator that appears on the screen has all the number buttons, the basic math functions (add, subtract, multiply, and divide), and even appears to have a little window at the top where the results of the calculations appear.

You use the onscreen calculator just as you would use a real calculator, by pressing the keys. You move the pointer over a number key on the calculator, say the number 5. Then you press the button on the mouse. The number 5 will instantly appear on the calculator's little window. Then you can move the mouse pointer over a calculator function key, let's say over the + (addition) key. Press the mouse button. Then press one or more number keys, say 99.25. Finally, you press the equal sign key (=), just like you would on a real calculator, to get the result. In this case, the answer, 104.25, will appear in the calculator's display window.

Like many of the other Macintosh programs, the calculator program is very easy to use. Why? Because you already know how to use it. Maybe the onscreen calculator is not the best way to use a computer to do math problems; until you get used to it, it's actually kind of tricky to get the mouse pointer centered exactly over the little keys. But punching the keys on a handheld calculator is something you probably already know how to do; most of us have done it hundreds of times. The learning transfer is great: It requires very little learning time, if any, to make the connection between pressing calculator keys with your fingers to pressing an onscreen representation of a calculator with a mouse pointer.

But, remember, that little window on the screen is not really a calculator. While this may seem obvious, the illusion is so complete that you might overlook some interesting factors. What you have really done when you select the calculator from the pull-down menu is to load a program. It is a program that basically does three things:

First, it loads a graphics program that creates the illusion of a calculator on the screen;
Second, it maps that area of the screen so that the appearance of a pointer can be coordinated with the illusion of a calculator key;
Third, if the mouse button is pressed when the pointer is hovering over a number key, a subroutine of the main calculator program is called to display that number on the screen;
Fourth, the program puts the appropriate math functions into operation.

This last part of the program's capability should suggest that the math program can be operated separately from the graphics aspects of the program (the mouse pointer
coordinated with the illusion of the calculator keys). And, in fact, it can. Notice one thing about the math function keys on the calculator: The multiplication and division symbols are not the ones traditionally found on a calculator. Instead, the Macintosh calculator program uses symbols that can be found on the keyboard: the asterisk for multiplication and the slash for division. These symbols have long been used on computers for these math functions and Apple has maintained them on the Macintosh calculator program. So if the +, −, /, and * symbols can be used for the four basic math functions, why not just enter them using the number keys at the top of the keyboard or the ones on the numeric keypad? You can use the math functions inherent in the calculator program without even using the calculator; just enter the numbers and one of the four math symbols from the keyboard. Interestingly, the numbers entered and the results of the calculations still appear on the calculator's screen, just as if you were still using the mouse to press the calculator's keys.

So if it's actually more convenient to use the number keys on the keyboard, why bother with the fancy graphics of the calculator program in the first place? The answer lies in the user interface concept we discussed earlier. The graphic, onscreen calculator gives the user access to the math program. The calculator provides a way to begin using the math functions of the computer right away. This concept is true of the Macintosh in general: The powerful new Apple software gives the user access to the capabilities of the machine—the shortcuts that require a little more knowledge are there when the user is ready for them.

**Alarm Clock**

Another interesting program is the alarm clock program. This program is also loaded from the pull-down menu labeled with the little apple symbol. And, like the calculator, it can be positioned anywhere on the screen (or even mostly off the screen). The clock should be accurate once it is set because there is a battery inside the Macintosh that keeps the clock on time even when the computer is switched off (or even if it is unplugged).

To reset the clock, click the mouse pointer at the right side of the box. This will reveal a lower section that shows a clock, a calendar, and an alarm clock. Click on the one you want to set, then click on the number of the time or date you want to change. Then you click on the up or down arrows at the right side to make the numbers smaller or larger. You can also set the clock from the control panel. When the alarm clock is turned on, don’t expect a jangling bell or buzzer that could wake you up from a sound sleep; this alarm clock has a single bell tone, designed to remind someone working at the computer when a certain time has arrived (time to knock off?). The alarm, once set, will work even if the clock display is no longer on the screen.

**The Note Pad**

The Note Pad is just what it sounds like, a place to type in and record quick notes. This is not a full-fledged word processor—it’s not meant to be. It’s just for making a few notes that you want to remember. However, once you have the notes entered and recorded, of course you can expand them into a larger document later. You can enter up to eight pages
of text on the small note pad, clicking on the lower left edge of the pad to turn to the next page. Notes entered into the Note Pad can be saved on the disk and/or printed out to an attached printer.

**The Puzzle**

The Puzzle is designed to mimic one of those little plastic games with sliding tiles. The idea is to arrange the numbers in sequence. This game is mindless (relaxing to some, irritating to others) in its original plastic form and is no less so when it appears on the Macintosh screen. The user can move the “tiles” by dragging with the mouse. We think the Puzzle is a kind of tongue-in-cheek joke that the designers of the Mac threw in not so much as a time waster (while the boss is away, the computer workers will play), but as a demonstration program that utilizes the computer’s graphics capability along with the mouse. The game is so simple that the boss should worry more about users playing with the Macintosh operating programs and the other desk accessories: They are far more interesting.

**Key Caps**

Key Caps is a special way to show you the various keyboard symbols available. After selecting the Key Caps program from the menu, press the Shift key or the Options key to get a display of all the symbols that the Mac can use.

**Control Panel**

The trickiest of the special onscreen programs—and the most interesting looking—is the control panel. Selecting the control panel from the pull-down apple menu actually selects a group of special programs combined in one complex display. As part of the control panel, there are programs that allow you to:

- set or change the time of day
- alter some of the display features
- set the speed of double clicking for the mouse
- change the pattern of the background on the desktop
- alter the speed of the mouse pointer’s movement
- set the volume of the speaker
- set the keys to repeat when held down, set the repeat rate, and set how quickly the repeating begins after the keystrokes

Let’s look at each of these mini programs within the master control-panel program.

At the top of the panel is the clock. This is a similar clock program to the one discussed above. Setting the clock from the control panel also resets the one in the clock program.

Just to the right of the clock display is a display that looks sort of like a set of stacked boxes, and to the right of that, some boxed numbers, 0,1,2, and 3. This is an odd
little program the controls the "shimmering" associated with the menu items as you select them. When you click on a menu item choice, that menu entry flashes a few times before the Mac goes looking for that item. Believe it or not, with this control panel program you can specify how much of the flashing takes place.

Just below that is a panel to control the rate of character flashing, from one to three flashes per second.

Down one more panel, in the corner of the control panel, is a display that allows you to set how fast you use the double clicking feature of the mouse. There are three choices: The left choice indicates slow double clicking (allows a longer wait between clicks) and the middle and right choices require faster clicks.

At the bottom center of the control panel is one of its most interesting features, a way to set the desktop background display. Notice the two squares with the dot patterns in them. These squares represent the patterns that can be used as the background for the desktop that appears on the screen whenever the operating program is loaded. The box on the left is an extreme closeup representation of the basic dot pattern. It represents only a very small area. The appearance of the desktop screen is the result of the constant repetition of this pattern across the screen. The box on the right is the larger view of that pattern. To see the many patterns that are available, move the mouse pointer to the upper part of the right box and click once. This will display the next available pattern. Keep clicking to see many more possible patterns.

If you want to make up your own pattern, just move the mouse pointer into the left box and start clicking: each click inserts a new dot into the pattern. To make sweeping pattern changes, hold down the mouse button and drag across the left box.

This may seem only an interesting exercise with no realworld value. But it can be useful. We use this control panel feature to customize the desktop dot pattern of each of our many diskettes. That way, we can tell which disk we are using as soon as it is loaded. It makes it a lot easier to tell if we have just loaded our master MacPaint disk or the one with the partly completed drawing (the MacPaint program takes up so much disk space it requires a new diskette after you complete a few drawings).
To the left of that display is one used to change the speed of the mouse pointer. Click the number on box on the left and the pointer will zip across the screen with very little movement of the mouse on the tabletop. Click the zero box and the pointer will not move so far when you move the mouse.

Along the left side of the control panel is the volume control slider. The slider demonstrates some of the remarkable graphics capability of the Macintosh, as well as its easy-to-use design. To change the volume of the Macintosh’s internal speaker, all you have to do is position the pointer over the slider and move it up or down while holding down the mouse button. When you let go of the button, the speaker will sound once to show you the new volume level. The program is designed to allow you to adjust a volume level that appears to have infinite gradations from soft to loud. And it lets you do it with an apparent sliding switch that may already be familiar to you; thus the Macintosh claim that you already know how to use many of its features. And although the sliding switch might seem simple, the graphics program that makes the visual image is not. The coordination between the screen display and the moving pointer must be maintained while the mouse is rolled across the tabletop. The processing required to alter the speaker’s volume and then activate it each time the visual slider is moved is no simple task either.

In the center of the control panel’s display is a way to reset the keyboard keystroke delay rate. The top row of numbers (zero to four) is flanked by a turtle and a hare. Use the mouse pointer and click a number to set the rate at which the keys repeat when you hold them down. Set it at zero and the keys won’t repeat at all, even if you continually hold them down. This might be the best setting for a beginner just making the transition from a typewriter to a computer; they tend to hold down the keys too long and end up confused when actions are constantly repeated. Set it to three or four for fast, very skilled computer typists.

The second line is to set the delay time before the key starts repeating after you press it. Again, set it at a lower number for beginners and increase it as the user grows more skilled.

**THE FUTURE**

And what of the future? Will other computers adopt the techniques of the Macintosh operating system? Will they place an easy-to-use shell around their operating systems? There are two answers: yes and yes. Yes, they will, and yes, they already have. Every few weeks now, somebody comes out with an easier-to-use shell program. Some of them are full-fledged operating systems. The only drawback right now is that no other computer maker has made the move that Apple did; that is, no other computer maker has come out with a very powerful hardware system sporting an operating system with an easy-to-use interface for unskilled users. That means that even the best designed of the new shells are saddling the hardware systems with a lot of new overhead. The modern operating systems are running on the outdated hardware systems and that means they are going to run slow.

For the moment, the Apple Macintosh and its advanced operating system is out in front of the pack. Others are pursuing, but it will take a while for anyone to catch up.
5

A CLOSER LOOK: MICROSOFT BASIC

Although there are several versions of the BASIC programming language used on today's personal computers, the version of BASIC developed by the Microsoft Corporation is generally considered to be the standard for the industry. It is probably for that reason that Apple decided that the first version of BASIC made available for the Macintosh would be the version from Microsoft. These days, it is almost mandatory that every new personal computer come equipped with a good BASIC language interpreter, and this version of BASIC from Microsoft is the most common choice. Still, it might seem surprising to some that Apple wouldn't stay with its tried and true Applesoft BASIC. But if you look closely at the list of early software that was released for the Mac, you will notice that the Microsoft set of programs is prominent. Apple worked closely with Microsoft to develop a set of well-rounded programs that would give the new Macintosh computer a running start. And BASIC is one of the key elements of that set.

MACINTOSH BASIC

Even though the Microsoft version of BASIC utilized on the Macintosh is supposed to be the same one used on other PCs, there are some differences, both in the command structure and in implementation. For one thing, since the Mac doesn't support color (at this time), the color-related commands are not implemented. A few of the graphics commands
also act differently, mostly because of the more advanced graphics capabilities of the Mac.

The big difference, of course, is in the user interface. All of the Macintosh ease-of-use procedures have been incorporated into the BASIC interpreter program. The program is loaded by clicking on the BASIC icon, or on any of the programs previously stored on the diskette. If you double click on a stored program, the interpreter is automatically loaded first.

Another difference in the Macintosh's particular version of BASIC is in the use of windows. Instead of requiring the user to enter every command, the program operation commands—for example, the RUN and LIST commands—can be used separately in individualized windows. This results in a windowed screen that looks nothing like the usual BASIC screen. Notice that the LIST window can be maintained on screen—even while the program is test run. Commands can be entered in still another window. All of the usual Macintosh windowing features such as scrolling, dragging, window sizing, etc. are in place. The result is a highly useful program that takes full advantage of both the very complete version of BASIC from Microsoft and the power of the Macintosh.

**USING "INCOMPATIBLE" BASIC PROGRAMS**

By using the "standard version" of Microsoft BASIC, Apple has hedged one of its bets in going up against the IBM standard. Since this version of BASIC is the same one used on the IBM PC and its many compatibles, Apple has made the Macintosh able to run any of the thousands upon thousands of BASIC programs that have already been written using Microsoft BASIC. Any code written for other computers using Microsoft BASIC can be run on the Mac, usually with very little modification. It's only a matter of either reentering the program on the Macintosh's smaller diskettes, or (even easier) using a modem to send the code as an ASCII file to the Macintosh.

Even if BASIC programs are written specifically for the IBM PC or any other computer, many of them can be successfully run on the Macintosh. BASIC is a universal programming language and is quite transportable; that means it should run on any computer that has a BASIC interpreter program. But to run a BASIC program on a computer other than the one it was intended for may require a few tricks. To get the program to run, you have to figure out generally how the programmer created the BASIC files, and specifically how the programs were intended to be loaded. The trick is to first determine how the loading sequence was originally written and then find a way to get around it.

You can even use programs written for other computers. But, you have to be sure that the program is written in a form of BASIC at least somewhat compatible with Microsoft BASIC. Otherwise, you may have to go into the programs and rewrite some of the BASIC code. Programs written in BASIC and intended for use with the IBM PC should be compatible with this form of Microsoft's BASIC. However, it doesn't necessarily work the other way around. Programs written especially for another computer may use a different form of BASIC, or the programs may call for special files to be found on that system's DOS diskette. Sometimes the only way to tell is to try it out.
Next, you have to determine the loading method of the BASIC programs. A common method of loading BASIC files on the IBM PC is to use an "autoexec.bat" file. This type of file automatically loads the various component files of a complicated BASIC program. This method can cause problems for the Macintosh. When the batch file sends the system off looking for the IBM version of BASIC, it is bound to get lost. The trick is to simply get rid of that "autoexec.bat" method of loading and load the program after you get into the BASIC interpreter.

The best way to find out if the program is going to work is to run it. BASIC has a way of checking for problems in the program as it runs. If you get an error that says something like "break in 350," it probably means the program has a statement in line 350 that is incompatible with the form of BASIC used by the Macintosh. Usually, if the program runs at all, these kinds of errors can be fixed. But it requires some knowledge of programming in BASIC to do it. Try reading the BASIC manual that came with the Macintosh BASIC program to see if the statement that is baffling the computer uses the same form in the book. Usually the problem will be in one of the graphics commands, like the SCREEN command. This means that the Macintosh program is not quite compatible with the way the original program handles its graphics screens. Sometimes, you can change the SCREEN command and it will clean up a lot of other problems in the program. Other BASIC commands such as COLOR, CIRCLE, DRAW, PALLETTE, and POKE can also give a computer fits if they do not handle these actions in the right way. A little rewriting can fix the problem and teach you a lot about BASIC at the same time.

HOW USEFUL IS BASIC?

To anyone not very familiar with programming languages, the obvious question is: what can I do with it? Actually using a programming language may seem like something best left to professional programmers. But we disagree. We feel the need to say this clearly:

DESPITE WHAT YOU MAY HAVE HEARD ABOUT THE PROS AND CONS OF PROGRAMMING LANGUAGES, WE BELIEVE THAT ANYONE IN THE BUSINESS OF TEACHING WITH AND ABOUT COMPUTERS SHOULD HAVE A WORKING KNOWLEDGE OF A PROGRAMMING LANGUAGE.

As logical as that sounds to us, we realize that it is a controversial statement. There are those that say it is the user interface with computers that is important, not the professional, or programmer, interface. Others will say that programming is dead, that authoring languages—often utilizing plain English commands—are taking the place of programming languages. But we say that even if you never write a computer program, even if you never teach a class on how to program, you still need to be familiar with programming: It is integral to a teacher’s knowledge of the history of computing, the function of the computer, and the awareness of the structure of all existing computer programs. It is the teacher who feels the master of the advanced concepts that can best teach the beginner the first important steps.

We know that there are a lot of successful teachers out there who don’t know much
about programming and who are still doing one heck of a good teaching job—and a lot of
them are using computers of one type or another as part of their lesson plans. Also, a lot of
parents have introduced their children to the wonderful new world of computers knowing
little more than how to turn the machine on. All this is possible because of the many
excellent educational programs now on the market. All you have to do is load one of these
programs and the computer takes over the teaching task. A skilled teacher can even en­
hance these programs by how they are managed in the classroom. But we believe most
teachers and parents could do even better if they knew how to program the computer. If
they could learn to control the computer by using some programming commands (the
commands in BASIC, for example), they could design their own teaching programs. We
are not saying the average teacher should develop programming skills to match the profes­
sionals, but we are saying that with such a powerful tool sitting there on the table, why not
learn to gain some control over it?

LEARNING A PROGRAMMING LANGUAGE

As you may have noticed, we didn’t say a teacher should learn any particular
programming language. They all, in essence, do the same thing. LOGO, for example, is a
popular language for teaching programming concepts to young children. LOGO is best at
controlling graphics displays on the computer screen, but few people realize that beyond
its graphics capability it is a capable programming language. Other languages you may
have heard of are COBOL, a programming language especially valuable for business ap­
lications, or FORTRAN, a language most useful for doing mathematical jobs (which is
known as number crunching in computer circles). If you want to learn more about the
other programming languages available for the Macintosh, we discuss several of them and
their use in the school education chapter.

And if these other languages are best for specific applications, what is BASIC best
for? We believe it is best for teaching. BASIC—which stands for Beginner’s All-purpose
Symbolic Instruction Code—was originally created for students. It may or may not be the
best overall programming language, but it is surely one of the easiest of the popular lan­
guages to learn. And it is different from some of the other languages in that each com­
mand is acted upon, line by line, in exactly the same order as they are entered. This means
that when you write a program, it is easy to keep track of just where in the program things
should happen. If you make an error, it is easy to discover where the error is by tracking
through the steps sequentially. And if you forget to tell the computer to do something, you
can just go back and insert a command at the appropriate place.

Some readers may be thinking that this is a job better left to the professional pro­
grammers that create commercial learning programs. Would it surprise you to know that
many of the learning programs you can buy today are written in BASIC? And that you
could write ones like them with just a few hours of training? It’s true. Many of those
programs are much simpler than they seem if you know the commands to control the com­
puter. In this chapter we present some of the simpler BASIC commands. With just those
few commands you can write your own elementary learning programs. By the time you
finish this chapter, you will be even able to write your own easy programs.
The Basic BASIC: Tryout

Ready? First let us tell you what a computer program is. It is nothing more than a series of instructions. The key thing to remember is the word series. You create a computer program by entering command words (instructions to the computer) in a series of numbered lines, one after the other. The following one line of code is a complete computer program:

10 PRINT "'HELLO THERE.'"

This one-line program demonstrates one of the nicest things about BASIC: Programs written in BASIC don't have to be of any special length and they don't have to include any special commands. A BASIC program can be one line long or 10,000 lines long (and some are). If we tell the computer to run our little one-line program it will begin with the lowest line number it finds (in this case the line numbered 10) and carry out the command (the first word in the line) as directed. It will print on the screen

HELLO THERE.

And then, finding no more lines numbered higher than 10, it will end. (Notice that the command PRINT means to print everything that follows the command, as long as it is enclosed in quote marks.)

But if we add one more line, like this:

10 PRINT "'HELLO THERE.'"
20 PRINT "'BYE BYE . . . .'

We now have an entirely new program of two lines. Since we haven't told the computer to go off and do anything complicated, anything that requires more lines of commands, we still have a complete program. Now if we tell the computer to run this program (just by typing in the word RUN) it will display this:

HELLO THERE.
BYE BYE . . . .

After this screen display, the program will stop. It detects the absence of any higher numbered line and automatically ends the program.
Notice in the tryout above that the computer displayed exactly the letters and spaces that were included inside the quote marks. Programming in BASIC is that simple; you just follow three main steps for each line of code:

1. Put in a line number
2. Type the command
3. Add any information the computer needs in order to carry out that command.

Even though some of the other commands are much more complicated, the principle represented by these three steps is always the same.

**SOME BASIC COMMANDS**

BASIC uses a large number of commands to control the operation of the computer. There are commands to control what appears on the Macintosh’s screen, commands to display the full of the computer’s graphics capability. There are even commands that will make the computer play music (the Macintosh has a built-in speaker capable of a wide range of sounds). Here are some of the most common commands.

**CLS**  
This stands for CLEAR SCREEN. If you enter this command, the computer’s screen will immediately be wiped clean. But don’t worry, the lines you have already entered won’t be lost. Every line has been saved in the computer’s internal memory.

**DATA**  
This command is used to create a list of data that can be used later by your program.

**DEL**  
This stands for DELETE. You use it to delete lines from your program if you change your mind. All you have to do is type

```
DEL 130
```

if you want to delete line number 130 from your program. This command automatically wipes out the line numbered 130 from the computer’s memory. A note here: This doesn’t mean you have 130 lines in your program; since you create your own numbering system, you might only have one line in your entire program, in this case numbered 130.

**DIM**  
This stands for DIMENSION. It is used to set aside, in advance, some memory space that may be needed later by your program. But don’t worry if you don’t know about computer memory and all that: If you don’t specify it, the computer will find its own memory area and it will automatically remember where it is. Later all you have to do is say go get it; the computer will do the rest.
DRAW

This is a fun one. With this command you can draw your own figures on the screen. This is another one of those commands that sounds very complicated to use, but really isn't. It's also one of the favorite commands of the pros who make pretty pictures on the screen. Here is the secret: All you have to do is tell the computer in what direction you want to move and how far. Then off it will go, drawing a line just as your ordered. Add several of these DRAW commands together and you end up with a computer-drawn picture. Learning to use this one command can give you a lot of the power of the LOGO programming language. You could write your own LOGO-like program and impress the heck out of your friends.

EDIT

This command is used to call up a line that you want to make changes in.

END

Use this command to tell the computer that this is the end of your program. It tells the computer to go ahead and do its final housekeeping, such as closing down all the open files and storing active memory areas; then it tells the computer to take you back to the BASIC system.

FILES

This is a very handy command. It is used to tell the system to give you a listing of all the files you currently have stored on the diskette. This is a way to check to see what is on the diskette without leaving the BASIC system.

FOR . . .NEXT

These are actually two commands used together. The FOR command tells the system that this is the beginning of a series of instructions that you want the computer to repeat several times. The NEXT command tells the computer that it has reached the end of the series and it is time to go back to the beginning, indicated by the FOR command. These commands are very useful. They allow you to tell the computer how to do some very tedious things that you would normally have to repeat over and over. For example, if you are having the computer draw a series of designs on the screen, this command can be used to tell the computer how many times to repeat the design.

GOSUB . . .RETURN

This is another combination of commands that tells the system to go off and find a series of instructions, called subroutines, and then, after it has finished, return exactly to this point and begin again to carry out the sequence of instructions.

GOTO

This command is used often. The GOTO or some form of it is part of every type of programming language. It is used to transfer control to a specified line number somewhere in your program. After your program finishes with a task you put one of these GOTO statements on the next line to tell the computer where to look for the next instruction.
IF . . . THEN

This is another one of those two-command combinations. It is used a lot like it sounds: IF such and such happens THEN go to the next task. This is a very valuable command when you are telling the computer to do some repetitive task until something happens. Let's say you have written a program to teach a young child how to use the keyboard. You have the computer ask the student to press the spacebar. All you have to do is put in one of these IF . . . THEN statements saying IF the student presses the spacebar (you can write into your program a way to tell the computer when that happens) THEN PRINT "That's right! You just pressed the spacebar." (Remember how to use the PRINT command?)

INPUT

You will use the INPUT command in every learning program you write using BASIC because this is the computer's link to the student. What this command does is stop the program and wait for an INPUT from the keyboard. If you have your program ask the student a question, this is how the computer accepts the answer. This command can also be used for other types of input besides answers to questions. For example, let's say you have written a program that has a little elf run across the screen full of letters. (You could make the letters a colorful landscape.) You can tell the student to try to press the letter that the little elf is walking on top of. You can easily tell the system what letter the elf is on top of, and if the student's INPUT matches you could PRINT a congratulatory message on the screen. Using BASIC, you can even have the computer keep score. (Are you getting the idea of all the things a teacher can do using BASIC?)

LET

LET is used to assign a name or a value to a storage place in memory.

LIST

This command is used to print out on the screen all the lines you have written in your program.

LLIST

Same as LIST, except it prints the listing on a printer if you have one attached to the computer.

LOAD

This is the command you use to tell the system to go and find a stored program and LOAD it into active memory.

LOCATE

This is a very handy command if you are writing graphics programs. It tells the computer to move the cursor to a spot on the screen that you specify as part of the command. You enter the LOCATE command, then tell the computer what row and column you want to place the cursor. Since a lot of the other commands start drawing displays on the screen wherever the cursor is located, this is how you tell the computer where you want the next design to begin.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>This command, a file management tool, tells the system to change the name of a file you have stored on a diskette.</td>
</tr>
<tr>
<td>NEW</td>
<td>NEW is used to clear a new memory space so you can begin writing a new program.</td>
</tr>
<tr>
<td>PLAY</td>
<td>This is one of the best commands in the BASIC tool chest. It is used to play music! This command lets you tell the computer to play notes (including how loud and how long). You can even specify an exact note (and its sharp or flat), and you can play more than one note at a time to make the computer play chords.</td>
</tr>
<tr>
<td>PRINT</td>
<td>As we have already seen, this very useful command is used to tell the system to PRINT something on the display screen.</td>
</tr>
<tr>
<td>LPRINT</td>
<td>This is the same as PRINT except the display also gets printed out on a printer if you have one attached to your system.</td>
</tr>
<tr>
<td>READ</td>
<td>This command is used with the DATA command we described earlier. It is used to READ the data you stored in advance.</td>
</tr>
<tr>
<td>REM</td>
<td>Use this command (it stands for REMARK) as the first thing in a new line if you don’t want the computer to take any action. This is usually used to make notes to yourself.</td>
</tr>
<tr>
<td>RENUM</td>
<td>This stands for RENUMBER and is used to renumber the lines of your program.</td>
</tr>
<tr>
<td>RUN</td>
<td>This one is simple. It tells the computer to go ahead and RUN the lines of your program that it has stored in its temporary memory. You can use this command even before you have finished your program to get a look at how it’s going so far, or just to check for errors. (One of the nicest things about BASIC is that it will automatically tell you if you have made a mistake. All you have to do is tell the computer to RUN the lines it has stored to check each line for errors. It will also check the logic of the entire program you are building.</td>
</tr>
<tr>
<td>SAVE</td>
<td>This is how you save a program on diskette.</td>
</tr>
<tr>
<td>SOUND</td>
<td>This command is used to tell the computer to make sounds. It is a little different from the PLAY command (you can specify the exact frequency of sound and the duration), but it can also be used to send sound through your home speaker system.</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>Last of all, this command is used when you want to get out of BASIC and go back to the DOS system.</td>
</tr>
</tbody>
</table>
There are many commands in the BASIC command set that are not included in the above list. These are just some of the most common ones. To use these commands and the others not listed, you should read the BASIC instructions that come with the Mac.

WRITE YOUR OWN PROGRAMS

We hope you began to get an idea of the programming power hidden in the BASIC language. The teacher or parent who gains control of that power will make the most effective use of the computer's full capabilities. All you have to do is sit down in front of the computer and open any one of the hundreds of books on using BASIC that are for sale today. There are probably ten available in your public or school library. And one of the best things about learning BASIC on the Macintosh is that the version of Microsoft BASIC that it uses is fast becoming the standard BASIC; you will be able to use your new programming skills on almost any of dozens of different small computers that are sold today. And just because BASIC is not an expensive, compiled language, don't make the mistake of thinking it is of limited usefulness. The enhanced form of BASIC provided with the Macintosh is all you should need to design programs as complete and as useful as a learning situation could require.
A CLOSER LOOK: COMMUNICATIONS AND MODEMS

If you read magazine and newspaper stories about computers, you have already heard it: The big thing in the future of computers is communications. And when you talk about computer communications, you are talking about modems. Modems are the devices that allow computers to talk to each other over the phone lines.

A modem—using a communications software package—translates the type of data a computer normally uses into a form that can be sent over ordinary phone lines. A modem at the other end of the phone can decode those signals back into computer-readable form.

This chapter takes a closer look at the growing field of computer communications, specifically examining the potential of using modems with the Macintosh computer. Despite the vast potential of the evolving technology high-speed data transmission (at present mostly involving large computers), we will restrict our discussion to the publicly accessible sources of information and individual-to-individual communications best suited to today’s small computers.

The future of computer communications will undoubtedly belong to high-speed, telephone-based voice and data transmissions; eventually we will all be hooked into the computer communications system we now call the telephone system. Telephone companies will work with our computers to handle our mail, our investments, our travel arrangements, our medical needs, and other aspects of our lives that we now think of as having nothing to do with technology. Our computers will tap into one central source that will retrieve any information we ask for; the computer will be our interface with business,
recreation, and education. Today, private computer communication is used by only a handful of computer users. But they are the pioneers. They are discovering the problems and solutions that will lead to the communication standards of tomorrow. Already, there are unmanned computers that can recognize voice commands. The technology will be refined, the applications will be broadened.

And education will be among the big winners. Today, educational computing is mostly limited to small systems and individual computers. But the power and capabilities of those small systems will grow. One unpredictable new development—such as networking, or perhaps some new advance in multiple-user communications systems—will lead to rapid new developments. Program and data sharing may be a first step toward new software developments that will lead to new teaching methods. Central to such data sharing schemes will be the communications terminals. Some believe computer technology will pass over education, as television technology did in the last decades. They point to the dominant control that commercial ventures have gained over television transmissions, and the resulting low quality of programming. But it seems unlikely that private capital can so dominate computer communications, simply because it is a two-way medium: Anyone can transmit. And there will be a lot of us out there listening.

The way to begin communicating is by buying a modem and tapping into the information base now being shared by others. Twenty-four hours a day, computer users are talking to each other, sharing data on almost every topic on earth. You get to choose. You only get what you want; you decide what you want and you ask for it. And, best of all, when you have something to say, you can send it out to the listening world. Let’s start there, with buying your modem.

BUYING A MODEM

Right now, buying a modem for the Macintosh is easy: So far, there are only a few available. But that is changing every day. By the time you read these words you will probably discover that buying a modem is becoming almost as difficult as buying a computer. As more computers are sold, more manufacturers begin to design and sell modems to go with them. There are now a large number of modems on the market. It takes very little conversion to make them work on the Macintosh. Probably every modem maker will produce a Macintosh modem and making a choice will become more and more like guesswork.

To help you understand how modems work and to decide on the features you may want, we have put together a list of some criteria on which to base your choice. Let’s look at each category and its implications.

MODEM STANDARDS

There is one fact of life about modem standards: In order to make our computers talk to other computers, we need them. Since modems talk to other modems, they cannot be
purchased as if they were lawn mowers. If your neighbor’s lawn mower is a rotary mower and yours is a reel type, and you want to swap, it doesn’t really matter much to either one of you. Both types do the job—cutting grass—equally well. But with modems, both yours and your neighbor’s modems must use exactly the same communications-standard operating method—usually referred to as communications protocols. If they don’t, your computers won’t be able to talk to each other.

Technical Note: About Modems

A modem (an abbreviation of modulator-demodulator) is a device used to change the kind of data that is used by computers (digital signals) into a form that can be transmitted over the phone lines (analog signals). Another modem is required at the receiving end to convert the signals back again. The main reason for such data conversations is that although a computer is capable of parallel data transfer, today’s telephone communication system is not. This means that even though data within a computer can be transferred across its internal devices in parallel—all bits in a string (a block of bits) being transferred at the same time—telephone equipment requires serial transmission. In serial transmission, bits are handled one at a time, in sequence. Since there are no logical blocks in serial transmission, the communications sending equipment must break the data down into packets, frequently inserting information into the string to tell the receiving equipment how many bits will comprise the next string sent; in other words, start/stop information must be sent along with the raw data telling the receiver how to reconstruct the transferred packets. Some understanding of all this may help explain why a modem is needed, why special rates of transmission must be specified, and why communications are so much slower than internal computer operations.

As modem technology was developing, there were a variety of modem protocols. Such protocols have always been necessary because the method of sending information down a wire one bit at a time means that there has to be some coding of the data. For example, start and stop bits have to be added to the data to indicate logical groupings of data. It is the sending modem’s task to guide the insertion of these codes so the receiving computer can decode one group of bits and display it. For example, the letter ‘‘A’’ is sent as a short group of signals. But there must also be a code sent to tell the receiving computer where the beginning of the ‘‘A’’ string begins and ends and then another code to indicate the beginning of the next coded string. It is the rules and organization of this coding that is meant when we talk of modem protocols.

In the early days of modem communications, a number of protocols coexisted. It was usually large businesses that bought modems, and they used them to allow two or more of their remote computers to talk to each other. The two computers were usually of the same type, so the modems needed only to guarantee that communication between the two identical machines would never be altered; any protocol that did the job was good enough once it had been decided upon.
But the tremendous growth of the microcomputer industry changed all that. Everybody was selling a different brand and type of computer, each with a recommended brand and type of modem. There was something of a guessing game going on wherein you had to try to predict, in advance, who you would be communicating with. Then, if you found out what type of communication standard was used by those computers, you could decide on that particular standard. The situation was a lot like that brought about by the diverse types of personal computers that were sold, except that when modems were finally standardized, there were very few holdouts.

What happened in the United States was that the all-dominating marketing force in telephone communications business, AT&T, decided to create a standard for modems used on its lines. AT&T couldn't force everybody else to go along with its new standard, but that wasn't necessary. All AT&T had to do was start making modems through its Western Electric subsidiary and attaching them to telephone equipment. The standard protocols used for these modems soon became the modem standard.

Today, two of AT&T's standards—the Bell 103 standard and the Bell 212A standard—have become the main communications standards for all personal computers. The Bell 103 protocol is used only for the slower speed transmissions, the 300-baud rate (more about modem speeds in the next section). The Bell 212A standard is more popular than the 103. This is because a 212A modem supports many different speeds. It can operate at 300 baud like the Bell 103 modems, but it can also operate at several slower speeds. And it has one other added feature: It can handle communications between computers at one much higher speed, the 1200-baud rate. And if you are using the auto answer feature of your modem, it can detect and answer in either 300 or 1200 baud, depending on the rate of the incoming call.

Other countries have also established some modem standards. But while in the United States our standards have evolved through the process of free enterprise, the Europeans have created their protocols in what they consider to be a more civilized manner—with a treaty. The treaty group (called CCITT) has established a standard that is very like the Bell 212A standard, called the V.22 protocol. But you can't use your Bell 212A modem to call up a European V.22 modem; the two can't talk to each other. The CCITT protocols are just different enough from the AT&T standards to make them completely incompatible. When (and if) that situation is going to be rectified is anyone's guess.

Anyhow, the choice of protocol for an owner of the Macintosh is fairly easy: Make sure it is either a Bell 103 or Bell 212A modem.

---

**SPEED**

Unfortunately, the buy or not-to-buy decision is not so clear when we look at the question of speed (rate of transmission). The speed of a modem refers to the maximum rate at which it transmits or receives data. It is measured in terms of units called bauds, which refer to the number of bits sent per second (bps). Modems with speeds of 600 bps or less are considered low speed; 1200 bps to 9600 bps are medium speed; and over 9600 is high speed. To give you an idea of these various speeds, consider that a 300 bps modem can
communicate about 30 characters per second. That means that you can send or receive about 250 average-sized words per minute. Now compare that with the amount you can send or receive using a 1200-baud modem—four times as much, or 1,000 words per minute. With the 300-baud modem you can transfer about forty pages from a book in one hour’s worth of connect time. But with the 1200-baud modem, you can probably transfer the whole book, 160 pages!

Still, fast is not always better. On occasion you may not want to store the information you receive. Maybe you just want to look at it. Words appearing on your computer’s screen at 300 bps are readable. Whole sentences appearing at the 1200 bps rate are rushing by too fast to read.

But let’s look at the overall costs. If you buy a low-speed (300-baud) modem for your computer, you are going to save some money on the initial purchase. How much? Well, because the 300-baud modem offer a lower technology option as compared to the 0 to 1200 baud modems, you get get one for $75 to $300. Compare this to the $500 or more you will have to pay to get a faster modem. You have to ask yourself whether you will be using the faster transmission capabilities of the 1200 baud modem enough to justify the higher costs.

There are some other considerations. Ignoring for the moment matters related to your personality (are you the patient type who can enjoy watching characters pay onto your screen at the slow 300-baud rate?), let’s look at how you will be using the modem. Will you be using it to contact some of the big, commercial databases? If so, they all support the higher speed, 1200-baud modems. And don’t forget the relationship between modem speed and “connect time.” These commercial information sources charge by how long you are connected. With a faster modem, you don’t have to be connected for so long. This is an important consideration. Let’s say you are in communication with a commercial database that charges you $10 an hour for the time you are connected. Let’s say that in one hour you can access 40 or so text pages of information from the database, using your 300-baud modem. This means that when your bill comes from the service you can calculate that it cost you about 2½ cents per page that you transferred. If, on the other hand, you used a 1200-baud modem, you could have transferred the same amount of text in one quarter the time, or at a cost of about 6/10ths of a cent per page. So it all depends on how often you plan to use such services. If you plan to use them often, it wouldn’t take long to pay for the difference in price of the faster modem.

But now that we have reasoned that out so carefully, we have to tell you that if you use certain database services there is a gotcha. Some commercial database services have begun to charge more for users of the faster modems. While it may seem unfair to charge higher connect costs to those who have already paid extra for a faster modem (at least it seems unfair to us), we must remember that these information sources are moneymaking concerns. They have to make a profit to stay in existence. And as faster modems became more popular, people were getting the information they needed faster and therefore not staying connected for as long; the companies saw their profits going down. So, in order not to penalize the people that were using slower speed modems by just raising all their prices, they only raised the price for the users of the fast modems. The difference is usually something like ten to fifteen dollars per hour: A typical example might be $5.00 per hour using a 300-bps modem (for non-prime-time, late at night) and $17.50 on a 1200-bps
modem. Still, the $17.50 is not quite four times the $5.00. So, in the long run, if you are going to be transferring large numbers of files and using up a lot of connect time, you might still end up with a considerably lower bill by using the faster modem. Again, you have to determine what your primary use of the modem will be.

COMMUNICATIONS SOFTWARE

If the hardware (the modem) is the mechanical link that allows communication with other computers, it is the communications software program that makes the process accessible to you. The modem provides the physical link between the computer and the telephone lines; the communications program sets up the coding system by which two computers can talk to each other. These programs guide the user through the complicated steps required to link up with another computer. A number of communications programs are sold these days, each of which does about the same kinds of tasks. But how the procedure is presented to you, the user, is what determines a communications program's overall effectiveness.

Modems are usually sold with communications software, but we found that some of the programs sold for use with modems were quite complicated to use. Some popular types of modems offer many advanced features, but the software can be somewhat hard to get used to. The best communications software programs use helpful screen messages that bring the new user along step by step and yet allow the more advanced user to move along quickly. Part of the problem with the software programs for such advanced modems is the many commands they require the user to learn. But it is the documentation's command-oriented approach and lack of helpful onscreen help routines that make using the programs cumbersome. With better-written program manuals and onscreen menus and help routines, even the most complicated modems can be accessible to all users.

Features

Another way to decide which modem to buy is to analyze the features that each provides. Modems only require the basic modulation and demodulation hardware to do their job. But there are a variety of other features that can make it a lot easier to use the modem. Below is a list of such features and an indication of the purpose of each. Decide which would be useful for the kind of communications you plan to do and then use them as a guide when you go modem shopping. First the hardware features.

Line Monitor Speaker

The line monitor speaker allows you to hear the telephone dialing sounds so that you can determine what is going on after the call is placed. Some modems use the speaker in the computer itself, while others use a speaker located on the modem board. Others use screen messages to indicate the status of a call. Each of these are effective, and it is basically a matter of whether you want to hear it or see it.
**Full or Half Duplex**

Full and half duplex transmission is really very simple. Half duplex communications allow only one computer to talk at a time. This means that once you are in contact with another computer you will only be able to transmit when the other computer is waiting. This can be a problem if you accidentally ask the computer at the other end of the line to display a file that is much bigger than you had expected. And what if it turns out to be the wrong file completely? There will be no way to interrupt the other computer to say stop. Full duplex transmissions, on the other hand, allow both computers to be on the line at once. You can get a message through to the other computer even while it is transmitting.

**Connect Signal LEDs**

These LEDs (light-emitting diodes) are often provided on the front of a modem to indicate the status of your transmissions. Sometimes, when you are not getting the messages you expect, it is difficult to tell exactly what the problem is. You may have been accidentally disconnected or just temporarily interrupted; or maybe there is a software problem. These little lights will help you decipher it.

**SOFTWARE FEATURES**

Just as important to your communications plans is the software package that goes with your modem. You can buy the package recommended for the modem you choose, or you can buy one of the stand-alone programs. Either way, the salesperson should be able to tell you if the program has the features listed below. Best of all is to try it out. See if the basic approach offered by the program is clear to you. Although you will have to learn all the commands peculiar to that program, there are various ways used by such software to present the command structure to you.

**Help Command Files**

These are helpful explanations of the frequently used commands and protocols used by the program. Are they easily accessible when you need them? Does the program respond to your requests with specific help for the task you were attempting? Are the help messages understandable, or do they confuse you even more? If you get a chance to try the program out, your initial response may be your best indication of how well the program uses this feature.

**Abort Command**

This feature allows you to stop a call before it is completed, should you change your mind or have an unforeseen interruption. This feature is handy, but not essential: You can always turn the computer off and start over.
Dialing Directory

This allows you to store frequently used telephone numbers. Usually, if the program has this feature, the program will also let you automatically dial a number just by entering its label.

Manual Dial

This allows you to dial a telephone number from the keyboard.

Redial Last Number

This feature lets you to redial the last number you called.

Number Linking

This allows you to specify a second number to be dialed if the first number is busy or does not answer. This feature can be a very handy aid if you are checking a large group of numbers because it can be used to set up a sequence of numbers to be dialed.

Automatic Answer

This is a feature you must be sure to get if you want your computer to be able to answer calls from other computers. With this feature, you can set your computer to receive data even when you are not there to keep an eye on it. The way to use this feature is to create or buy a program that starts into action when the computer receives a call.

Busy Mode

This allows you to take the phone off the hook to let computer callers know you are temporarily using your computer for other tasks. They will receive a busy signal.

Protocol Detect

This is another handy feature. It tells your system to detach the protocol of an incoming call and switch its own protocol to match that of the caller.

There are a number of other features, but we won't list them all here. Just be sure when you are investigating which modem to buy that you find out what features are available and if they will meet your special requirements.

SOME MODEM HISTORY

Only a few years ago, modems were considered devices used only by electrical engineers and professional programmers. They were a way to stay in touch with the big mainframe
computers (often called the *host* computer). Before the age of personal computing, all data and the software programs to handle that data were stored in these very large, very expensive computers. Just about the only people who got to use a computer were the people that worked for the businesses that owned them. If you wanted to use any computer software program or look at any of the stored data, you had to get in touch with some sort of timesharing software that gave you access to the central computer for a limited time. The usual way of doing that was to sit down in front of a *terminal*—sometimes called a *workstation*—and type in your special (and secret) password. The computing task was done inside the mainframe computer; the terminal was only a window into it, a display screen under the control of the host computer. Because these big computers were so very expensive, you almost always had to share time on the computer with many other people using other terminals, and that could lead to frustrating delays while the computer kept you waiting for your turn.

The original modems were a way to communicate with the host when you couldn't be in the main office where the terminals were. New terminals that were (sometimes) smaller and lighter than the office versions were designed with modem-like hardware built into them. These communications terminals could usually only communicate with one type of mainframe computer because the communications software was in the host computer: The terminal had only the hardware to make the physical connection, using the phone lines, and then the big mainframe computer took over, treating the remote terminal just as if it was one of the normal terminals in the main office. The remote terminal usually couldn't send or receive files for permanent storage. And it couldn't be contacted by any other computer. The remote terminal could be tied into a phone line—just as a modern modem-based computer can—but there the similarity ended. Today, we can attach a modem to our fully capable computers and call up almost any other type of computer there is. We can send text files or programs to the computer on the other end of the line or we can type in messages from the keyboard. These files can be automatically stored on disk at either end. And there need not be a supervising operator keeping watch over all this. Today's communications software can handle the sending and receiving processes all by itself—if you tell it to do so.

In a sense, the remote communications terminals were an early demonstration of the need for the personal computer: People found that they couldn't always be where they could make contact with the main computer, so why not make a small computer, with its own storage and software, and do the work elsewhere? When the processing was complete, the resulting files could be sent into the main computer via the phone lines. Or a small workstation could *downline load* (ask for files to be transferred to its local storage) where the user could use diskette-loaded programs to analyze that data without tying up the main computer. But that wasn't how it happened. The mainframe computers continued to be timesharing oriented, with numerous remote terminals. Meanwhile, the personal computer developed along its own course, rarely being used to communicate with the big host computers. Until now—demonstrating one of those paradoxes that seems to occur in any history—it is the personal computer people who are discovering the advantages of being able to contact a central source of information. Of course, the personal computer has a big advantage over the old communications terminals. It is, like the big mainframe,
a true computer. It has its own control program, its own storage areas, and now it can even have its own communications software. Instead of being treated like a dumb display window, it can send and receive files from any other computer, of any size. You can still use it to contact any of the big mainframe computers (if you can get their phone number), or you can contact your pal’s little home computer anywhere there is a phone line.

**REACH OUT**

Now that a lot of people are buying full-capability computers (with adequate memory, disk storage, industry-standard software, and modems), something new is happening to personal computing. Computer people are starting to make contact with each other. The modem attached to your computer makes it possible. If a computer is turned on and the communications software is operating, day or night all you need is that computer’s phone number; you can tell your computer to give it a call. You can even program your computer to make the call automatically, in the middle of the night, to save phone charges. You can contact your friend’s computer, leave a message or read one, ask for a file, or send a file. The communications software can handle it all.

**HOW DO I USE A MODEM?**

Although you can use your computer to call up your friends and type messages back and forth, that kind of communication is better done with voice transmission—in other words, by using the telephone in the good old-fashioned way. The real value of communicating with computers is to automatically transmit information. Endless pages of stored text can be transmitted over the phone lines, even if both of the human parties are sound asleep.

Once modems became widespread, it didn’t take long for people to begin sharing information. At first, about the only people who had much to share with the world were people who had something to sell. If you had your computer call in their computer, they would show you all the services you could buy. Later, the information built up to the extent that all they were selling was access to the information itself. This is still one of the main uses of computers with modems. You can access so many different kinds of information, with so many entries in each category, it would be impossible to list them on a *hardcopy* (that’s computer lingo for printing on paper). If you know the right phone numbers, you can get information about everything from the price of stocks on the New York Stock Exchange (to the Tokyo or Paris stock exchanges) or the best time of the year to buy Australian melons (in March). You can find out about the latest computer games or about next year’s football games; you can even buy advice on how to bet on either. Some of these services are very specialized, others have access to huge databases with just about every type of information under the sun.

The only problem with using your modem to access these databases is that the information is so complete and so interesting, it can cost you a lot of money to browse through it all. Remember, these commercial services charge by the amount of time you are on the
line (with some you may even have to pay phone charges, though most use a toll-free 800 number). The database service charge can add up to real money if you use it often enough. And some of these services have grown databases so large, it can take you quite a lot of expensive time on the line just while you are searching through the huge lists of categories. Here’s a tip. Most of the larger services provide an online training program to tell you about all the categories and to teach you how to use the base. Though you might be tempted to just start looking for the information you want, ask for the HELP program when you first sign on: It will save you time and money in the long run.

**Definition: Database**

A database is any kind of stored information that can be accessed by computers. A list of all of Russia's past premiers is a database; so is your grocery shopping list. If all of the words in the *Encyclopaedia Britannica* were to be stored in a computer file (they have been), it would constitute a database. But so would the text of a letter to your friend in Indianapolis.

Here are the names and numbers of some of the larger database services:

- **BRS**
  1200 Route 7
  Latham, NY 12110
  (800)833-4707
- **CompuServe**
  5000 Arlington Center Boulevard
  P.O. Box 20212
  Columbus, OH 43220
  (800)848-8199
- **Delphi General Videotex Corp.**
  3 Blackstone Street
  Cambridge, MA 02139
  (800)544-4005
- **Dialog Information Retrieval Service**
  3460 Hill View Avenue
  Palo Alto, CA 94303
  (800)227-5510
- **Dow Jones News Retrieval Service**
  P.O. Box 300
  Princeton, NJ 08540
  (800)222-0081
- **IBM**
  Systems Products Division
  P.O. Box 1328
  Boca Raton, FL 33432
  (800)447-4700
  (800)322-4400 (Illinois)
  (800)447-0890 (Alaska, Hawaii)
At the end of this chapter we provide a more complete list of consumer databases and a description of the sort of services they provide.

ABOUT BULLETIN BOARDS

As a result of the growing size of these general databases and their relatively high cost, some people got the idea that it would be cheaper and easier just to ask for information from others who were on the lines. So they began to leave their computers turned on all the time, giving out their phone numbers as "meeting places," where people could call in and ask questions. Then others could call in later with their answers. All it required was someone willing to let their computer serve as one of these "bulletin board" meeting places. There was no charge except for the price of the phone call. This is an idea that will grow fast now that modems and communications software are becoming more common. These bulletin boards have tended to specialize. You can find bulletin boards used for everything from dating to the exchange of information about sailing, science fiction, travel, trucks, and old trains. Of course, the majority of the information to be found in these bulletin boards is in one way or another about computers.

One of the best reasons for adding a modem to your Macintosh is the ever-growing number of programs that can be found on the bulletin boards that specialize in the Lisa and Macintosh. It seems as if many people who own a computer with the capabilities of the Macintosh love to learn more about how it works. And as soon as they figure something out, they are on the phone lines sharing that information with others. Programs that can be downline loaded (quickly transferred from the database to your diskettes) for free from other computers number in the thousands. Educational programs abound on these bulletin boards.

So how do you get the phone numbers of these bulletin boards? One answer is from other bulletin boards. This is not a problem once you are into the loop (as it is called). But how do you get into the loop; how do you first find the good numbers? One way to get started is to buy a copy of one of the many newsletters that often go along with bulletin boards. These newsletters are often not very professional; they are usually just a monthly mimeographed sheet, or maybe just an every-so-often one-pager. But that's as it should be. Bulletin boards are the people's way to share information.

We have a telecomputing newsletter, nothing fancy, but specializing in educational
computing; and we will send you a free copy if you send us your name and address and the type of computer you own or are considering buying. Here is the address to write to, as well as our 24-hour-a-day bulletin board phone number (our bulletin board also lists some of the other bulletin board numbers that specialize in educational computing):

American Systems Computer Institute (ASCI)
809 West 800 South
Salt Lake City, Utah 84104
Bulletin Board Phone #: (801)582-3128

It seems to us that despite the large number of educational programs out there, the number of bulletin boards that specialize in educational computing is surprisingly small. Maybe this could be your big chance to get involved. Perhaps you or your school would like to become a bulletin board meeting place for information about educational computing. All it takes is a computer with a fair amount of storage available and a phone line. If you decide you want to give it a try, write or call us and we'll give you some tips on how to get started. We'll also list your number on our educational computing bulletin board list. You may be surprised how much fun you can have, in addition to all the new things you can learn. Remember, it's the people that run the bulletin boards that have the most fun: They have access to all the information. If you run your own bulletin board, you can get a printout of every day's activity and go over the information with your group of educators or parents. It's an idea that any school should consider, especially if it is still in the process of acquiring educational computing equipment and/or software.

As promised, here is a more complete list of databases and online services.

**CONSUMER DATABASES**

BRS, Inc.
1200 Route 7
Latham, NY 12110
(800)833-4707

Includes information in following areas:

- Business and Financial
- Education
- Energy and Environment
- Reference
- Science and Medicine
- Social Science and Humanities

Compuserve, Inc.
5000 Arlington Center Boulevard
Columbus, OH 43220
(800)848-8990
Includes information in following areas:

<table>
<thead>
<tr>
<th>Advertising</th>
<th>Languages on Compuserve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advice</td>
<td>Manufacturers' Newsletters</td>
</tr>
<tr>
<td>Air Travel</td>
<td>Medical</td>
</tr>
<tr>
<td>Auto Information</td>
<td>Microcomputers</td>
</tr>
<tr>
<td>Autos, Buying</td>
<td>Movie Reviews</td>
</tr>
<tr>
<td>Aviation</td>
<td>New Product News</td>
</tr>
<tr>
<td>Banks</td>
<td>Newsletters</td>
</tr>
<tr>
<td>Book Ordering</td>
<td>Personal Finance</td>
</tr>
<tr>
<td>Budgeting, Home</td>
<td>Politics</td>
</tr>
<tr>
<td>Business News</td>
<td>Popular Science Magazine</td>
</tr>
<tr>
<td>CB Radio Simulation</td>
<td>Professional</td>
</tr>
<tr>
<td>Canadian</td>
<td>Radio</td>
</tr>
<tr>
<td>College Board, The</td>
<td>Real Estate Ads</td>
</tr>
<tr>
<td>Columbus Area</td>
<td>Sigs</td>
</tr>
<tr>
<td>Commentaries</td>
<td>Securities</td>
</tr>
<tr>
<td>Commodity Service</td>
<td>Shopping</td>
</tr>
<tr>
<td>Computer Club News</td>
<td>Software Reviews</td>
</tr>
<tr>
<td>Editorials</td>
<td>Special Interest Groups</td>
</tr>
<tr>
<td>Education</td>
<td>Sports News</td>
</tr>
<tr>
<td>Energy Conservation</td>
<td>Stocks, Bonds</td>
</tr>
<tr>
<td>Entertainment</td>
<td>Travel</td>
</tr>
<tr>
<td>Food Information</td>
<td>Travel, Abroad</td>
</tr>
<tr>
<td>For Sale</td>
<td>Trivia</td>
</tr>
<tr>
<td>Games, Computer</td>
<td>USDA Standards</td>
</tr>
<tr>
<td>Gold</td>
<td>U.S. News</td>
</tr>
<tr>
<td>Golf</td>
<td>Washington D.C. Area</td>
</tr>
<tr>
<td>Home Management Programs</td>
<td>Weather</td>
</tr>
<tr>
<td>Index</td>
<td>World News</td>
</tr>
<tr>
<td>Investments</td>
<td></td>
</tr>
</tbody>
</table>

Dialog Information
Services Inc.
3460 Hillview Avenue
Palo Alto, CA 94304
(800)227-1927

Includes information in the following areas:

<table>
<thead>
<tr>
<th>Agriculture and Nutrition</th>
<th>Law and Government</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bibliography—Books/Monographs</td>
<td>Materials Science</td>
</tr>
<tr>
<td>Business/Economics</td>
<td>Medicine and Biosciences</td>
</tr>
<tr>
<td>Chemistry</td>
<td>Multidisciplinary</td>
</tr>
<tr>
<td>Current Affairs</td>
<td>Online Training and Practice</td>
</tr>
</tbody>
</table>
Directories
Education
Energy and Environment
Foundations and Grants

Patents and Trademarks
Science and Technology
Social Sciences & Humanities

The Source
Source Telecomputing Corp.
1616 Anderson Road
McLean, VA 22102
(800)336-3366

Includes information in the following areas:

Business Services
Communications
Computations
Consumer Services
Education
Entertainment
Publishing
Travel Services
News and Information Services

EDUCATIONAL DATABASES

Directory of Online Information Resources
CSG Press
11301 Rockville Pike
Kensington, MD 20895
(301)881-9400

CSG Press puts out a directory twice a year, which although not online itself, provides a guide to more than 350 online databases. CSG can also access to most of the other databases and can perform data searches.

DataSpan
c/o Karl Zinn
Center for Research on Learning and Training (CRLT)
109 E. Madison Street
Ann Arbor, MI 48104
(313)763-4410

This project is funded by the National Science Foundation and is particularly interested in computer applications for science and mathematics educators. They provide information in the following areas:
The Dialog information retrieval service has been providing information to libraries for the past ten years. Information available includes:

- Computers
- Electronics
- Exceptional Child Education Resources
- Physics
- National Foundations/Funding Sources
- Selecting Computer Systems
- Using Local Communications Networks

The Educational Resources Information Center (ERIC) provides information from educational research. It is composed of clearinghouses across the country that abstract, index, store, and disseminate information about different topics in education. The clearinghouses listed below are directly involved with information on computers in the classroom.

ERIC Clearinghouse on Information Resources
Syracuse University
School of Education
130 Huntington Hall
Syracuse, NY 13210
(315)423-3640

ERIC Clearinghouse on Elementary and Early Childhood Education
University of Illinois
College of Education
Urbana, IL 61801
(217)333-1386

ERIC Processing and Referencing Facility
4833 Rugby Avenue, 303
Bethesda, MD 20814
(301)656-9723
Now that we have looked at the hardware, the operating system, and the many capabilities of the Macintosh, we want to discuss how the Macintosh can be put to work in your home. We will discuss ways that you can use your computer to enhance your children’s education, to help you manage the family budget, to compute your taxes, to write correspondence or speeches, to keep a journal, to organize lists, and to communicate with data banks around the country.

Any personal computer is a great asset to a home business or to the administration of a civic group or organization, but the Macintosh is particularly valuable because of its ease of use and its advanced graphics capabilities. In addition to keeping your accounts straight using one of the many database programs, or writing correspondence using the MacWrite program, you can create illustrations and charts that will improve your communication. You can add that personal touch that makes a lasting impression. And you can do all this with very little training or advance reading.

If you listen when people talk about the Macintosh, they almost always mention something about how easy it is to use. Computers are intimidating to many people. They fear making fools of themselves, breaking the machine, or getting stuck and not knowing what to do next. The Macintosh puts these fears to rest. It was made for the person who is afraid. Every design decision was made to facilitate ease of use. Once you learn how to use the mouse and a few basic rules about the pull-down menus, you’re on your way. From that point on you just keep experimenting, learning more and more as you go. Then if you want to read the instructions, you feel like you know what the writers are talking
about. Instead of plowing through a lot of dense documentation, wondering if you’ll ever understand, you can use the book as a reference to solve specific problems.

Although we intend to provide further suggestions for using your computer as a business machine, what we are mostly concerned with in this chapter is how to use the Macintosh to advance your children’s education. As we examine the possibilities we will discuss such topics as finding the best place to keep the Macintosh, working out a timesharing agreement between you and the kids, using the computer for a variety of educational and family experiences, getting involved in community and school computer projects, and using the computer to continue your own education.

**FINDING THE BEST PLACE TO KEEP THE MACINTOSH**

One of the first decisions you will have to make if you purchase a personal computer for your home is where to put it. This is an important question since it can determine how much the computer will be used by the various members of your family. Too often, computers end up in the back of a closet after the excitement has worn off. It’s just too much trouble to pull it out every time you think of using it. Although the Macintosh is lightweight and can easily be moved from one spot to the next, think about setting up a work area that is accessible and where there is plenty of storage to keep operating manuals, disk, ribbons, paper, and software packages. If you want your children to use it for education put it somewhere that is comfortable for them. Make sure there is a chair that allows them to view the screen and use the mouse and keyboard without straining. And make sure it is a pleasant environment.

The room where you put the computer should be relatively dust free since a computer generates static electricity and can be adversely affected by it. A room without carpet would be best, or you can spray your carpet with a static-reducing chemical. Also, it’s important to have a three-pronged grounded electrical socket. You may want to go to the computer store and purchase a special plug that is designed to take care of power surges. A change in the current can result in the loss of a file or program you have been working on if you haven’t stored it recently. It can even result in damage to your computer and/or to printers and other devices attached to it.

If you really want your children to use the computer for learning, maybe do some programming or study for their college entrance exams, select a quiet room where they won’t be distracted by what the rest of the family is doing. When making this decision you should consider the age of the children who will be using the computer, what they will be using it for, and the spot that best meets the needs of all members of the family.

**THE FIRST ENCOUNTER**

Let’s suppose you have decided where to put the Macintosh. You have it assembled and ready to go. If you have little experience with computers, Apple can provide a special learning program that is combined with a tape recording. It’s called the “Guided Tour of
MacPaint and MacWrite.'" Your cassette tape recorder plugs right into the back of the Macintosh, and the program will show you how to use the mouse, how to open menus and make selections in the MacPaint and Macwrite programs. You will be able to see on screen what is being described on the tape and the whole family can quickly learn how to make the Macintosh do its stuff.

You shouldn't be afraid to let young children experiment with the computer. Teach them to be gentle with the mouse and not to pound on the keyboard, and then let them try out a few things on their own. MacPaint provides an exciting sketch board for kids. They can draw pictures, erase mistakes and make changes, shrink or enlarge their drawings and finally, print them out so they can be hung on the wall or the refrigerator door.

WHO'S TEACHING WHOM?

If on the first day the new computer arrives you sit down in front of it for the next three hours while the kids are supposed to wait patiently and observe, you may be starting out with a problem. The attention span of young children is very short and they become bored quickly. By putting them in the position of looking over your shoulder you are likely to convince them that the new computer is really something for the adults; they will quickly lose interest. Certainly a timesharing arrangement will have to be worked out later on so that you can get your turn on the computer, but initially it is very important to allow the children to develop an affectionate relationship with it.

Actually, it might be more productive to have the kids show you how to operate the computer, since they probably know more about it than you do. They've probably used them at school or at a friend's house, or at the very least they have played some of the computerized video games. Let's face it, kids are growing up in a computer age and take computers for granted. The good news is that with the Macintosh it won't be long before you are computing on the same level as your children, and maybe if you work hard you might be able to show them a thing or two.

COMPUTERS AND CONFIDENCE

It is interesting that kids who have been using computers at school frequently take on a new role in the family when a computer arrives. The fact that they have something they can teach their parents gives them an increased sense of power and confidence within the family structure. They have something of value to contribute. This is an unexpected benefit of bringing the computer into the home. Take advantage of this phenomenon and use it to help your child develop a healthy attitude about his or her value to the family.

WORKING OUT A TIMESHARING PLAN

There is always going to be one child in the family who will take to the computer faster than the others. Because this child is more skillful, he or she will probably want to domi-
nate the machine, and the other kids may tend to shy away from the computer and leave it to him or her. If you want to be sure each child gets equal time, we’ve got a suggestion that might work.

Remember when we talked about the phenomena of children who have been using computers in school taking on a new role in the family when a computer arrives? This is where you can put that phenomenon to work. Or rather, put the child to work. Encourage the child to teach the younger brothers and sisters. Of course you’ll have to get across the notion right away that a teacher should instruct and then let his or her students try the lessons on their own, without too much supervision. Maybe the older child can write programs for the younger kids that will help them out with their studies, or even teach them how to program. If you work it right, the computer can be central to the development of strong and positive relationships between the various members of your family.

One thing you may find is that once everyone in the family gets hooked on the Macintosh, you may have to start saving up for another one. This is particularly true if the adults are going to keep the computer occupied a good deal of the time for business or if one of you is working on the Great American Novel. Once you get going with this computer you can spend the whole day clicking away with the mouse while the rest of the family members wait impatiently for you to finish. Parents who see the computer as a working tool may be unwilling to turn it over to young children. But it is just as important for children to have their time with the computer. If there are no computers in their school this is doubly important. How each family solves this problem is an individual matter. It is not always possible for a family to buy more than one computer. The Macintosh is not an inexpensive machine. But you should be aware that real difficulties can arise if you don’t have a plan developed for timesharing.

**CREATING A HOME CURRICULUM**

If you want to keep in touch with what your children are doing with the Macintosh and learn a few things about computing at the same time, you can set up a weekly family meeting where all the members get together and show something they have been working on. The children might want to pass out copies of a drawing they made using MacPaint, or a story they wrote using MacWrite. Maybe one of the kids has written a program that he or she can try out on the rest of the family. These presentations will get progressively more complex as the children learn to use more and more of the computer’s capabilities.

Make these weekly meetings a special occasion for everyone. Have a snack and sit around talking computers. Find out what’s going on at school. If the kids have computers in the classroom, you can talk about what kind of programs would be helpful for supplementing the school curriculum. Or if a child is having difficulty in a particular area of school work, you can discuss what kind of computer activities would help him or her to catch up with the other students.

To put the computer in a larger context, you could have one of the children do some research and you could all talk about where and how computers are being used in society—in the work place, supermarkets, banks, manufacturing, laboratories, medicine, NASA, and all the rest.
HOME EDUCATION

In a speech on September 27, 1982, Secretary of Education T.H. Bell issued a warning to educational professionals: "The information explosion is bringing a new trend to the job market of our country, a new trend to our entire culture. It will be national suicide if education continues to drag its feet in this area because if we do, the rest of the world will run right over us. . . . As educators we owe students the opportunity to gain the knowledge and skills that will make them employable. The future is going to mean literally millions of new jobs in computers and related technological fields."

In addition to this call for training in computers and related fields, there has been much talk about the decline of education in this country. Educators are being blamed for allowing mediocrity to creep into the schools. Perhaps this decline in the quality of education is as much due to the technological and cultural changes that Secretary Bell refers to as it is to a decline in educational methods or incompetent teaching. If we are truly changing from the industrial age to the age of information, then it is obvious that the skills required to operate and keep society running smoothly are changing also. It is obvious that teachers who have been trained according to old methods are not going to be able to teach the new skills without some retraining. How can a teacher teach computer programming if he or she knows nothing about it?

If parents are concerned about the quality of their children's education and believe that computers should be an essential part of the curriculum, then they must get involved in helping the schools get the financing for equipment and in convincing educators that they had better get themselves in for some retraining.

Many people today are suggesting that as more and more families purchase microcomputers much of the education of our children will take place in the home. Computers are appearing in the schools in greater numbers, but, as we have already suggested, it is a complex and time-consuming process to train teachers in computer skills and to restructure the traditional curriculum to include their use. It takes planning to decide who gets to use the computers, when, and for what purpose. It is much easier for a parent to purchase a computer, bring it home, and put it to immediate use towards their children's education. Though the cost might seem a bit steep, a lot of money is already being spent each year on educational toys and equipment by American families for their children. If you think about it, a computer is as good a place as any to spend these educational dollars.

Even when computers are present at school there are seldom enough to go around. In 1982 the Department of Education estimated that only one-third of the nation's public schools had one or more computers. At that time there was an average of one computer for every 400 students enrolled in this country's public and private schools and colleges. So until the schools get up to speed it is obvious that if you want your child to gain an expertise with computers, you will have to provide one at home. At the same time, get involved with your PTA, work with teachers and school administrators—let them know you're willing to support them in whatever way is necessary to integrate computers into the school.

Meanwhile, think about the advantage you are giving your children by bringing
home a personal computer. Not only are you making them more competitive in school, increasing their versatility and advancing their position in the job market of the future, but you are allowing them an opportunity to be responsible for their own education. At the same time you are putting yourself in a position to become involved in that educational process. How often have you heard yourself or other parents complain about the quality of your children's education? The microcomputer offers you a chance to do something about it. Not to take over the full responsibility for your child's education, but to supplement and fill in the gaps that cannot be covered by the school system for whatever reason. Once you are knowledgeable about these machines and the educational possibilities they offer, you will be in a better position to work with the school system towards integrating computers into the curriculum, and developing a cooperative relationship that will give your child the best education possible.

The fact that parents are interested in getting computers into the school is evident from the January 1983 issue of *PTA Today*, which was devoted entirely to urging local PTA groups to support the role of computers in the school curriculum. It was suggested that PTA-owned computers be installed in the schools if the funds to purchase them were not available elsewhere. The concern was raised that with the present influx of computers into all areas of society, if one's child's school does not have computers and there isn't one at home, one's kids will have a hard time competing with those who do have computer access. Many people are worried that the gap between those who are knowledgeable about computers and those who are not may be as serious as the gap between the literate and the illiterate. It is up to parents and educators to make sure that all children get an opportunity to learn the computer skills they will need.

**PERSONAL COMPUTERS AND YOUR PRESCHOOLER**

In this section we're going to talk about your preschooler and the Macintosh. We'll find out what experts are saying about young children using microcomputers.

When you were deciding whether or not to purchase a home computer, you probably didn't have your preschooler in mind as the primary user, but there is no reason why a child can't be introduced to the computer even as early as age three or four. If you give careful instructions and watch them for the first few times, even four-year olds can insert the Macintosh disks and operate the equipment without damaging it. Young children get excited about computers. There's something that intrigues them about pressing keys on the keyboard and making something happen on the screen. The response is immediate and this teaches children that they are in control of the machine and that it is a tool for their use. With adults there are all kinds of phobias that affect the interaction with a computer and make it a painful experience, but kids who are introduced to computers at a very young age grow up accepting computers as a familiar part of their environment. This is just exactly the response we want if our children are going to be adaptive to the technological revolution that has taken hold of our society.
YOU DON'T HAVE TO READ TO USE IT

With the mouse-drive Macintosh, children are able to interact with the computer long before they can read. But don’t forget the keyboard. It can be used to teach children the letters of the alphabet. Get into MacWrite and select the 24-point letter size of your favorite font (choose one that is legible). Show your child how to find the letters on the keys. It’s exciting for kids when they know that they are making the letters appear on the screen. Through repetition children will begin to make the association between what happens on the screen and the particular keys they have pressed. In addition to learning the letters of the alphabet children are learning to use the computer keyboard. Soon they will be writing words and simple sentences and learning how to touch-type. If you’re a hot programmer you can even write a program that calls up the traditional “A” is for apple and “B” is for ball.

In addition to learning the letters of the alphabet, young children can learn to count using the numbers at the top of the Macintosh keyboard. Certainly, these are things that the child would learn anyway from books, toys, or by counting their own fingers and toes. But there is something more involved here. Each interaction that children have with a computer helps to build confidence and a healthy attitude towards the machine. Also, by pressing the keys and making something happen, children are taking the first step toward understanding how a computer works. They are learning ideas such as “INPUT” and “OUTPUT,” but without the labels. Just as certain toys provide basic models for later mechanical and mathematical thinking, the computer, even when used for simple children’s games and educational programs, introduces fundamental ideas that will be invaluable for full mastery of the computer’s capabilities later on.

A BRIDGE BETWEEN A BOOK AND TOY

Dr. Richard Hefter, who is the author/illustrator of more than 150 children’s books and who has recently become involved in creating educational programs, sees the computer as a “bridge between printed materials and the toy concept.”* What he is referring to is that children can physically manipulate a computer in the same way they can manipulate a toy. There is an interactive element that is not present when a child simply reads a book or looks at the pictures. The child can make something happen. The “active” component gives the child a sense of power and confidence and also aids in the development of logical thinking. Dr. Hefter is in no way suggesting that we give up our children’s books; rather he is pointing out that interactive capability of computers so frequently discussed as a positive aspect of educational computing.

ABSTRACT TO CONCRETE

Dr. Ann Piestrup, of The Learning Company, sees the computer as another kind of bridge—a bridge to abstraction. She says, “The computer reacts immediately when the

child touches the keys... it turns the abstract into the concrete. And when children realize they cause the action, it builds the best foundation for being comfortable with the computer."

In an article entitled "A Computer in the Nursery School," which appears in Dale Peterson's book *Intelligent Schoolhouse*, Piestrup writes about the computer as being in the same category as clay, sand, fingerpaints, and building blocks, in that "all can be touched and easily manipulated by the child." She suggests that "a well-programmed nursery school computer can be: something to touch, hear and see.*

A program for children should be something that is playful, and that inspires the child's spontaneity. It should be aesthetically appealing, using color, music, animation, and form to keep the child's interest and engage his or her fantasy. And it should be immediately responsive to the child's "message" since long delays will cause the child to lose interest. It is very important that these programs are never threatening or judgemental. We do not want our children to become afraid of the computer, but to learn to like it very much. Another thing to keep in mind is that a program that is too difficult will not teach the child anything. There should be room to grow in the program, but only as the child is ready.

**EDUCATIONAL SOFTWARE FOR PRESCHOOLERS**

Although there is not a large number of educational programs available for the Macintosh at this time, they will be appearing very soon. In the appendix we provide a list of what was available at the time this book went to press. Check with the software publishers listed to find out if they have produced addition programs. Once a publisher writes one program for a particular computer it is likely that they will come out with others. You should also check with the dealer where you purchased your computer and ask for a list of all available software. There are a number of magazines that review software for the Macintosh, including *MacWorld* and *St. Mac*.

In addition to the magazines that deal specifically with the Macintosh, there are a growing number of books on the market that discuss ways to use microcomputers with your children, what's being done in the schools, the philosophy and theory of microcomputers in education, and the effect of computers on society as a whole. You will find a list of some of the books we have found useful in Appendix A.

**START AN EDUCATIONAL USER'S GROUP**

If you want to share ideas with others and find out what they are doing with microcomputers, you can join one of the users groups that are popping up all over the country. There are probably several in your area. If there isn't one close by, you can start your own group, and rather than gearing it to a particular microcomputer you can focus on

education. You can meet weekly or monthly to share ideas about what software you’ve found particularly valuable for your kids or get help with questions that have come up regarding the operation of your computer. You might even want to get involved writing and sharing programs. These are the kinds of groups that will be bringing our communities up-to-date and helping us find our way into the “information age.” The whole area of educational computing is wide open, just waiting for creative people to step in and develop it to its full potential. There is no reason in the world why you shouldn’t be one of those who are making things happen at the leading edge.

One of the first things you might want to do with your user group is to put together an educational software library. It might be part of the community library, something you set up with the school, or just a spare room in someone’s house. Of course it would be best if everyone had compatible computers so that all the programs could be shared. But even if you had a couple of different kinds of computers, if all of you put in some money to purchase programs everyone would have access to more programs than could be afforded individually.

BECOME AN ADVOCATE FOR COMPUTER LITERACY IN YOUR COMMUNITY

If you become excited about computers and want to share your excitement with other members of your community there is a book put out by Reston Publishing Company, Inc. called _Computer Town: Bringing Computer Literacy to Your Community,*_ which is based on research sponsored by the National Science Foundation on how to start a grassroots computer literacy project. The authors, Ramon Zamora, Julie Anton, and Liza Loop, offer ideas and suggestions for organizing events to introduce computers to your community. They review a number of projects they were involved in, presenting successes as well as failures. Read the book and if you want to get involved, you can even become an official Computer Town affiliate, and a member of the network of over 100 public access, community-based, computer literacy sites around the U.S. and the world.

EDUCATIONAL CONSIDERATIONS FOR OLDER CHILDREN

There are many ways that computers can be used to improve the education of your older children. If they are having problems in a particular area of school, say math or science, you can select tutorials or simulation programs which will help them catch up with the rest of the class. With computer programs they can move along at their own speed. This is particularly valuable for children with high IQ’s who tend to get bored in the school class-

*Ramon Zamora, Julie Anton, and Liza Loop, “‘Computertown: Bringing Computer Literacy to Your Community,’” Copyright 1983 by The People’s Computer Company, P.O. Box E, Menlo Park, CA 94025.
room since they already know what is being taught. If your child is already leaning towards a specialty and is eager to learn as much as he or she can, the computer can provide a tutor.

**ENCOURAGING MATH SKILLS IN GIRLS**

Some people believe that the reason girls have so much trouble with math and mechanical kinds of things is that they are not encouraged to play with mechanical devices, such as go-carts, bicycles, airplanes, engines, etc.; these provide a concrete basis for the abstract concepts of math and physics. Between the ages of eight and nine girls tend to turn away from studying science, math, and technology. Perhaps the early interaction with computers will provide a stimulating and nonthreatening environment in which girls can also learn the skills and concepts required to feel comfortable and competent in engineering, mathematics, and science.

When watching your daughters’ interaction with the computer you should be sure that they are getting equal time with the boys. Perhaps there are other reasons why girls turn away from math and science, but it is more likely that they just don’t get the same chance as boys to interact with tools that develop that kind of thinking. We will never know for sure until girls have the same kinds of early experiences as boys. The computer may provide an opportunity to find out why this is happening.

**TEACH THEM TO TYPE**

Even though we have suggested that children don’t have to be able to type in order to use a computer program, it is a good idea to teach them this skill as early as possible. It is too difficult to break bad habits of the hunt-and-peck method of typing once it has been learned. Your children will need good typing skills if they are to engage in such computer activities as programming and word processing.

There are a number of typing tutorials that will soon be available for the Macintosh, including Typing Tutor III from the Simon and Schuster Electronic Publishing Division, Type Attack from Sirius Software, Inc., and Typing Tutor from Southwestern Data Systems. These programs provide drill and practice to help children increase their speed and improve their accuracy. Even though these programs can teach typing skills, some teacher instruction may be required. If the school system doesn’t teach your children how to type before the age of eight or nine, you might want to get together with other parents who have purchased home computers for their children and see if you can get a class going.

**PROGRAMMING: A VALUABLE SKILL**

If you want your children to learn how to program you should be sure to purchase one of the many programming languages available for the Macintosh. Microsoft Corporation has
written a special BASIC for Mac. Other languages that are available, or will be soon, include, PASCAL, LOGO, COBOL, FORTH, FORTRAN, C and others.

Most personal computers run BASIC, which is a language that fairly young children can master. Since the language LOGO was developed particularly with young children in mind, you might want to see if it is available for the Macintosh. Digital Research Inc. should have Dr. Logo available by the time this book is published.

Another language, which is being used in high schools to prepare students for advance placement in computer science, is PASCAL. PASCAL is generally considered more advanced than BASIC, but it need not be seen that way if approached systematically. PASCAL uses the same general set of programming concepts as BASIC, but gives the programmer more control over the computer's screen graphics. There are also useful commands that allow you to interact more directly with inputs from the keyboard or other input devices. PASCAL is an especially useful language if you want to try your hand at writing game programs.

To help your children learn to program, you might want to purchase one of the books full of programs that can be copied into the computer. This will teach them how the various commands work and help them to become proficient at debugging and revising the programs they will write themselves.

There are also a number of tutorial programs (listed in the final chapter) that can teach them how to program with each language. Perhaps your children will come up with programs that can help you manage the family finances, compute taxes, or balance your checkbook. These all seem like minor things when they're rattled off like that. You will often hear people say, "Why do I need a computer to balance my checkbook when I can do it just fine with a calculator." They may have a point. But in this case, balancing your checkbook is really just an end product of the more important goal of teaching your children how to program. Also, those people who say they can balance the checkbook just as easily with a calculator don't know how nice it is to have a neat record of all their checks, outstanding and paid, organized by category and acceptable by the IRS at tax time. Of course you'll still have to have receipts or returned checks as proof of purchase. An added bonus is that all of your records are stored on disk rather than bits and scraps of paper that can get lost or crumpled up. If there are mistakes that need to be corrected, you make them on disk and your final printout will be neat and clean.

You can see by the discussion above that one aspect of computing leads to another. If someone writes a program, someone else can use that program. The activities that your children engage in with computers are productive as well as educational. If your son or daughter uses a prepackaged drill and practice or tutorial program, the product is knowledge about the subject that is being taught. But there are other computer activities that result in learning on a number of levels. As you have seen, programming is one of these activities. It teaches you both how to program and how to present information and solve a realworld problem such as balancing your checkbook. In using applications programs, students learn how to use the program (a word processor or an accounting or database program) and at the same time learn such practical things as how to write, how to think and organize ideas, how to do financial analysis or manage a small business. Let's look more closely at how this type of program can benefit your child's education.
A WORD PROCESSING PROGRAM
AS AN EDUCATIONAL TOOL

If you don’t know about word processing programs, you should. They are among the most useful software packages you can use on your computer. Had we been forced to use a conventional typewriter for this book, it would have taken many more months to write and hours of proofing and retyping. MacWrite is great for doing short papers, particularly if you want to illustrate them with some creations made with MacPaint. If you want to do some serious word processing, you will probably want to buy a more powerful word processing program such as Microsoft Corporation’s Microsoft Word or the Horizon Software Systems word processing program. Megahaus Corporation also has a program called Megawriter that should be available soon.

It is likely that when your children start using a word processor to prepare written assignments for school, their grades will improve. That is, if you have teachers who are open to the new technology. If they grade your children down for using computers or come right out and say papers written on computers will not be accepted, then it is probably time for you to go in and have a long talk with them. Certainly children must learn how to write the letters of the alphabet in the early grades, and we are not suggesting that typing on a computer should replace such skill development. However, once these skills are learned—and even at the same time they are being learned—there is no reason why children shouldn’t be able to take advantage of the organizational and editing options available on a word processor. If they think a paragraph might be written more clearly and don’t have to start from scratch each time a mistake is made, they are much more likely to get involved in the revision process which is so critical to well-organized and powerful written communication.

When the educational focus is all on the end product and trying to make the page look neat, there is less opportunity to explore ideas and be creative in the writing itself. Creativity is stifled; “blocks” often hamper even the most talented young writers. It is much more beneficial to focus on the process of writing and the development of imagination.

Writing is also central to developing skills for critical thinking. It is also important to the development of a strong sense of self that can be communicated to others.

People with highly developed communication skills have an advantage in all areas of social life. We would suggest that one of the first programs you purchase for your computer is a word processing program.

A word processor uses the display screen as a blank sheet of paper and the keyboard like any conventional typewriter, allowing the user to enter written text onto a diskette. Once the text has been entered (or as it’s being entered) changes can be made using such word processing features as “insert and delete,” whereby characters, words, lines, and even paragraphs can be deleted or moved around with the mouse. Other features include, “word wrap,” which allows the user to enter long paragraphs without ever hitting the return key. And with the “right-hand margin justification” feature, the computer makes sure that no word sticks out beyond the margin you set. Another feature in most word processing programs allows you to move whole blocks of text from one place to another,
inserting it right in the middle of another paragraph or any place you’ve decided you want it. You can even move it to another file.

In addition to these regular word processing features, you can get an additional utilities program that will check your text for spelling errors, using a built-in dictionary. There are even programs that tell you what kind of words you are using, whether they are active or passive, or if there is a tendency to use the same words again and again.

If you do purchase a word processing program for your children, you had also better purchase a printer, since without one there is no way to get the finished product.

**TRAINING FOR COLLEGE ENTRANCE EXAMS**

Although the people who create college entrance exams such as the SAT, GRE, and ACT say that you can’t train to take these tests, there is some evidence that practice can increase scores. Programs that help students train for these tests have been frequently reviewed by computer magazines and seem to be valuable tools. Most of them present a series of questions with multiple choice answers. When the student chooses an answer some programs go on to the next question, providing a score when all have been answered. Other programs give the student hints if the question has been answered incorrectly, helping the student to straighten out faulty thinking. These kinds of programs are particularly valuable in that, rather than just presenting questions similar to those which will be on the actual exam, they teach students how to think in ways required by the exam. They provide practice in the kind of logic that is needed to do well. Also, practicing and becoming comfortable with answering questions under pressure (most of these programs have a built-in clock) helps to alleviate some of the stress associated with taking examinations.

There are quite a few practice college entrance exam programs on the market, but we have not seen any that are ready for the Macintosh. However, since a number of universities are providing Macintoshes to their students through Apple’s University Consortium, we would expect that there will soon be Macintosh versions of the college entrance exam programs.

When making a decision on which one to buy be sure that you try the program out before you make the purchase. If it just presents one problem after the next without feedback it won’t be as helpful as one of the programs that give detailed help.

**MODEMS: A LINK TO THE WORLD**

In the chapter on modems, we looked at computer communications and suggested that this might be one of the most valuable assets of the computer. There are literally hundreds of databases online throughout the United States and other countries, and there are sure to be even more in the future. Many of these databases are small bulletin boards where information is shared between users with similar interests, but there are other large ones that have
huge data banks with information on anything from recipes to international weather reports, educational resources to athletic equipment. You can order books or get movie reviews. Get reports on medical science research or environmental issues.

You may not want to purchase a modem just yet, but keep it in mind as something to check out and maybe purchase in the future. You need to look realistically at the costs, including telephone bills, and service fees, and all the rest of it. The thing is, some people think that the modem is the piece of equipment that will truly bring education home. When students can hook up to the best libraries in the world and request information to be printed onto a disk or their desktop printer, or when a number of personal computers can be hooked together so that there is two-way communication between students and a teacher, then you begin to see the possibilities of home education. This may be a scenario from the future, but the technology is already available and waiting.

SUMMER COMPUTER CAMPS FOR KIDS

There is an interesting phenomenon popping up all around the country. A lot of schools and other organizations are starting computer camps or adding computing to the summer activities of long-established facilities. It seems that parents are interested enough in making sure their children get training on computers that they are willing to pay extra for adding several hours of computing lessons to children's summer activities. Even youth organizations such as the Girl and Boy scouts are getting involved.

When kids arrive at camp and find a cabin full of computers, the problem is not how to get them in there using the machines, but how to get them out. Children like computers. Generally they are required to work with computers just a few hours a day, and camp supervisors make sure they don't spend all of their time looking at the computer screen.

If you think it might be a good idea to send your children to one of these summer camps, check into them carefully and make sure the instructors really know how to use computers and that the lessons learned will be worthwhile. You don't want to spend extra money just so your kids can play computer games all afternoon. Find out what kinds of things will be taught, and that the plan fits in with your child's level of experience and knowledge. One thing you should watch out for. A number of computer stores offer so called "computer camps" that are actually little more than an opportunity for them to demonstrate their computers and try to sell you one. In the back of this book we have provided a list of computer camps and a source for getting more information.

In this chapter we have discussed ways to use your personal computer as an educational tool for your children. In no way is this meant to be comprehensive, since it would take many books to cover all the possibilities of educational computing. We have, however, provided a list of such books, and many other resources in Appendix B, and suggest that the chapter Education: The Macintosh at School is also valuable for parents.

One thing you should remember as you go through these chapters on education at home and school: Since adults are learning about computers starting at about the same level as most kids, much of what is suggested as applicable to children is also applicable
to adults. When you sit down with a young child to play an educational game, you are receiving valuable instruction for later computing you will do. If you help your child solve a word processing problem, you’ll be teaching yourself something you need to know at the same time. We hope that you will use the computer as an impetus for developing an atmosphere of equality in your family in which everyone shares and learns from each other so that learning becomes a way of life for all of you.
In this chapter we will look at some questions you should ask yourself before you start setting up a computerized classroom. These questions will help you with such things as planning a budget for initial and ongoing expenses, locating sources for funding, choosing the best educational software available, and developing a strategy for putting the computers to work as soon as they arrive. Although we are talking about how to use the Macintosh as a school computer, the subjects discussed apply to all personal computers.

Since needs vary from one classroom to the next depending on the age of the students and the subject being taught, we will look at a variety of classroom settings and discuss how a computer might be used in each of these settings. In planning your computerized classroom, you should consider the ideas discussed in this chapter and then expand upon them according to your own teaching objectives. If you can visualize the computers in your classroom before they actually arrive and begin to develop a plan for using them, you will be in a much better position to choose software and peripherals that will help you to meet your teaching objectives.

Later in this chapter we will take a look at educational software and give you an indepth guide for purchasing the best of what’s available. Remember, it’s the software that makes a computer into an educational tool.
QUESTIONS TO ASK YOURSELF

This chapter is presented as a series of questions. They are designed to get you thinking about a variety of issues that must be considered as you go about purchasing and designing your computerized classroom. Since the answers are interdependent, each question should be considered in the context of the others. When you have finished answering all the questions you will be ready to purchase your computers and put them to work.

WHY DO WE NEED COMPUTERS IN THE SCHOOL CLASSROOM?

This question should be considered on two levels. First, do we as educators have a responsibility to teach children about computers given their increasing importance in society, and secondly, can computers help teachers improve the quality of education in individual classrooms?

Is It Really Necessary To Be Computer Literate?

The answer to this question may be obvious to you, particularly if you have gone so far as to start reading books like this one on particular kinds of computers. However, it is still a big question for many people. As you go about looking for a computer, trying to find funding, and selecting software, you’ll run into problems if you don’t have a good answer to this question. In addition to feeling confident yourself that there is a need for computers in school, you may have to convince others if you expect their support, particularly if that support is financial. Also, if you understand why computers are valuable to education, you will be in a better position to select useful software and peripherals.

If you need a justification for bringing computers into the school there are significant social, economic, and political reasons to support the move. The computer, and particularly the microcomputer, is a tool that is moving into every area of society. But it is more than just a useful tool, it is a tool central to the development of a whole new way of doing things. We are moving from an economy based on the production of manufactured goods, “the industrial age” as it is called, to the “information age” where most of the country’s workers will be involved in the processing and distribution of information.

We have all thought about it, seen it in movies, or read about it in books. Computers are here and they’re bound to bring change. Your attitude about this change and about computers will depend on the point of view you take. You can see computers as an evil technology out to steal jobs from American workers, or as a technology that can take on the monotonous tasks humans prefer not to do. You can say that computers are impersonal and that they will be damaging to human emotions and relationships, or you can see them as valuable timesaving devices that can free people up to develop better relationships and engage in creative activities. You might be afraid that computers will be used to invade
your privacy. On the other hand, computers can be used to improve communication and promote the sharing of ideas between all peoples of the world.

The mythology surrounding computers is rich and varied. How you incorporate these myths into your own attitudes about computers is up to you, but as an educator you must take a broad view and consider what is best for your students.

**Some Objections You Might Encounter**

When we talk to educators about using computers to train their students for the future they sometimes say that, although they agree that many of the jobs of the future will require computer skills, most jobs will require very few skills at all, since they will be mainly service jobs that even a high school dropout can handle. They say that if students are required to take computer courses as part of their formal education, many of them will be frustrated when they can't find jobs that use their skills. Another objection often raised is that even if students do get jobs using computers, they will not need to know anything about programming or how a computer works. It will be just a matter of learning what buttons to push.

While we agree that not all students will be working at jobs that require computer skills or knowledge about programming, we do not see this as a reason for keeping computers out of the school. If computers are really at the center of a basic change in our society—the way we work and the way we interact—it is important to have knowledge about them, even if it is not required by our employers.

Even if you are not of the philosophy that schools should play a role in training students for jobs, there are good reasons for teaching children how to use computers. When used to their full potential, computers can be an instrument for incredible learning experiences. The computer can be used to learn about science, music, art, engineering, writing, economics, and almost any other subject you can think of. To say that students don’t need to know the high tech skills involved in computing, is to prevent them access to knowledge that can add to the quality of their lives. We do not learn just to make money.

Some people believe that learning computing skills such as programming develops something even more valuable. It teaches students how to think. If there is one thing schools should be teaching students to do, it is how to think. This along with helping the child to develop an intrinsic motivation for learning is really what education is all about. Computers help develop this intrinsic motivation for learning.

When students write a computer program or work with a commercial educational program, they are taking responsibility for their own learning. There is no teacher standing over them. The interaction is very personal and creative. Kids who have computers at home are in a particularly interesting position for intrinsically motivated learning. They are taking advantage of an opportunity to learn what they are most interested in, and at their own rate.

There is another reason to have computers at school. Some people believe that a
huge gap will grow between those students who have access to computers and those who do not. What happens when computers are purchased for schools with large budgets, but are not available for poorer districts? The discrepancy between the have and the have-nots could be as great as that between those who can read and write and those who can't. This is no reason to keep computers out of schools. It is critical that the government on the federal, state, and local levels get involved in providing computer equipment and teacher training for all schools. It is the only way children from poor families will be able to compete for high-level jobs and be accepted into many of the colleges and universities.

**Computers in the Colleges and Universities**

Another important reason that students in primary and secondary schools need to have experience with computers is that many of the nation's colleges and universities are adopting a policy that requires students to have computer skills. Some universities, are going so far as to require that incoming freshmen purchase a computer to use while they are in school. Other schools, although they do not require the purchase of a computer, still encourage students to buy and use computers by offering discounts or setting up computer labs around campus. Students with no computer experience may have trouble getting accepted into these schools.

We think that before long college entrance exams such as the SAT, MAT and the GRE will be given on computers. Although students should be able to take the tests without knowing how to operate a computer, having some familiarity with the machines would help to reduce anxiety. There are a number of programs that can help students study for these exams. We will talk about them in more depth later in this chapter.

**Can Computers Help Teachers Teach?**

Let's consider whether or not the computer is of value to the teacher and the educational process itself. There are a number of reasons why computers make good teaching machines. Listed below are some of the advantages computers have over other educational media.

1. **An Active Rather Than a Passive Tool** One of the advantages that frequently comes up when educators talk about computers in the classroom is the computer's interactive capability. Working with a computer allows the student to engage in a lesson actively; and educational psychologists agree that active learning is far more productive than passive learning.

   You can see the benefit of the interactive nature of computers if you watch a student working with a simulation program. With this type of program the student is actively involved making decisions that affect the direction the program will take. It is not a matter
of merely answering questions and being told if the answer is right or wrong. Different things happen on the screen depending on the student's interaction with the program.

Programming presents another opportunity for a highly interactive relationship with the computer. Not only must the student learn to communicate with the computer using one of the various programming languages, but he or she must learn to think logically. For instance, in writing an educational program, the student programmer must first decide what he or she wants to teach, then develop an overall strategy for presenting the information. Once the strategy has been developed the individual units of the lesson must be put in a logical sequence. This could be a sequence based upon level of difficulty, as in a drill and practice program, or on a hierarchy of prerequisite skills and knowledge as in a tutorial.

Careful examination of the various considerations involved in teaching someone something, teaches the student programmer to think about things that otherwise might be taken for granted. Even if the student is not writing an educational program—perhaps he or she is creating graphics or a game—logical thinking is still required in that the student must tell the computer what to do and what order to do it in.

The process of debugging a program after it is written is a very active process of trial and error. The motivation to get the program right is high because of the large amount of time it takes to create a program. If the student wants to work with or share his or her program it must first be debugged. There is a great sense of pride that comes from seeing a program you have written run all the way through.

With programming, learning takes place not only in the active process of the programming itself, but in the development of the content of the program and its presentation. If the program is to be useable, it must be structured logically, and therefore the programmer must have a user in mind and must think about how that user thinks. The student must consider the communication of information and how that information is received and processed, and how it will affect another individual. The development of these communication skills will be valuable to the student no matter what direction he or she takes upon leaving school.

2. AN INDIVIDUAL LEARNING EXPERIENCE Another advantage offered by the computer is that it can present individualized learning experiences based upon a child's unique capabilities. By keeping track of a student's performance, a computer program can adapt its presentation of a lesson according to the student's responses. If a student is having a great deal of difficulty, the program can branch to a set of easier questions or provide explanations that will allow the child to continue on with the present questions. In addition to adjusting to the student's level of progress, some programs have pretests and posttests that can help students determine at what level they should begin working.

The individualized computer lesson is particularly valuable for students who have difficulty following instructions and who interrupt the class again and again to find out what they should be doing. The fact that a computer program can continually instruct while it presents a lesson allows even slow children the chance to finish a lesson and develop a sense of pride about their ability to complete an assignment.

When students can move at their own pace, a higher level of interest is maintained
and the kind of motivation that is important to developing a sense of responsibility for one’s own education is developed. We all know that when students are self-motivated, learning takes place at an accelerated rate.

3. IMMEDIATE FEEDBACK One thing that you may not have considered about a computer program is that it can give the student immediate feedback even in a test situation. A testing program can be designed so that the student answers the question, the answer is recorded for grading, and students immediately find out if the answer is correct. Students are more likely to remember the correct answer if they have immediate feedback. Generally that information is not received until several days later when they are already involved in other activities. In addition to giving immediate feedback, a testing program can be designed so that it corrects faulty thinking. If a student makes the same kinds of mistakes again and again, the computer program can branch to an explanation that will set him or her on the right track. With these features, a test becomes a learning situation as well as a source of information helping the teacher determine the students progress.

4. SIMULATED LIFE EXPERIENCES We mentioned this feature of computers briefly in the section on interactive learning. But simulations offer more than an interactive learning experience. They give the student an opportunity to engage in “real life” situations that would not be available without computers. Although video and film can give access to a variety of worlds outside the classroom, the student cannot enter those worlds as an active participant. The student is not involved in what happens in the world of a film or video. As the technology of interactive video is perfected and becomes financially feasible, this area of computer programming will provide extraordinary opportunities for learning. Presently students can do such things as perform chemical experiments in a simulated science lab and participate in the decision making process of a large business; consider the possibilities of being able to make things happen with interactive videodisk or videotape.

WHAT DO WE WANT THE COMPUTER TO DO?

The answer to this question is important in deciding what hardware and software features you will need for your computerized classroom. The answer will vary greatly depending on the age of your students and the subject you are teaching.

Here is a list of some of the kinds of things you might use a computer for. Be sure to write down any ideas you have for using computers as you read this chapter.

Prepackaged Programs as Teaching Aids

There are four basic types of educational programs: drill and practice, tutorial, simulation, and educational game. While each of these types of programs have practical use in the classroom, some take better advantage of more of the computer’s unique capabilities. At
present there are not a lot of educational programs of any type for the Macintosh, but we are predicting that before long you will have a variety of programs to choose from.

In this section we provide a description of each type of educational program. We discuss educational applications and make recommendations for the type of hardware and peripherals you will need to take advantage of the programs you select.

**DRILL AND PRACTICE PROGRAMS**

**Description:** Drill and practice programs are designed and used to aid students in memorizing specific facts necessary to the development of spelling, vocabulary, and math skills. But they can also be used to help students practice answering any kind of question that requires a specific answer. This could include programs that answer questions about names and dates of important historical events, names of state capitals, French equivalents of English words, and the kinds of questions found on college entrance exams such as the ACT, SAT or GRE. The drill and practice format is also used in programmed tests. Teachers can create a large database of test questions from which individualized tests for each student can be generated.

When students begin working with a drill and practice program, they are generally asked questions to determine such things as subject category, number of questions desired, and level of difficulty. In some cases the program can determine the correct level of difficulty based upon records of the user's previous work on the program. These more sophisticated drill and practice programs can also provide information for the teacher to keep track of individual student progress. Programs that allow a new set of questions to be written in once the original material is learned will have a much longer life and keep up the students interest.

The most basic drill and practice programs present questions to the user either one at a time or in a column similar to a workbook format. The student answers each question and as long as that answer is correct, the program moves on to the next question. Some programs have no facility for getting a student out of trouble if they get hung up on a wrong answer, and will just continue to ask the same question again and again until the student gets it right or gives up. The better programs have the capability to either go on to another question (recording the mistake for later review) or to branch to some kind of explanation which will help the student to come up with the right answer. This branching feature allows the student to correct faulty thinking immediately and also makes the drill and practice process more interesting and productive.

There are some drill and practice programs that use color, graphics, sound, and animation to motivate and entertain the student during the practice session. Other programs use a game format that keeps students entertained while they learn. Although the Macintosh does not have color at present, the high resolution of the monitor presents such high quality graphics that it hardly matters.

**Educational Applications:** Drill and practice programs are valuable in the elementary school classroom for teaching vocabulary, spelling, math, names and dates, and other "facts." If they are well-written, they can take some of the drudgery out of the process of learning basic facts and skills that students in these early grades must learn as a basis for reading, writing, arithmetic, and logical thinking. These simple programs, particularly
those which incorporate a game format with sound and animation, are extremely valuable for introducing young children to the computer for the first time. They can learn something while they have fun.

Since drill and practice programs are the easiest kind of educational programs to write, you will probably find that there are more of these than any other type of program. Make sure that you answer the question, "What do I want a drill and practice program to do?" Then, select a program that meets your needs, referring to the section in this book on how to buy educational software to make sure you get a high quality program. If you are working with very young children, make sure there are plenty of graphics and other motivators to maintain their interest. This is less important with older children; still, there is no reason why learning shouldn't be fun. Keep this in mind when making your selections.

Testing: We will discuss testing in a later section in this chapter; here we just want to point out one thing. A drill and practice program without any frills most resembles a multiple choice, true and false, or fill-in-the blank, test. Keep this in mind when you create tests for your students. You can use the drill and practice programs that you have purchased to give you ideas for making your tests more interesting.

College Exam Preparation: Sometimes the only way to learn something is to practice it again and again until you can see it with your eyes closed. That's the way it is when studying for college entrance exams such as the ACT, SAT, and GRE. Although the organizations that create these tests say that there is no way your students can study to improve their scores—"they either have it or they don't"—there is some evidence that students can at least learn how to take the test. They can learn things such as how much time they should spend on each question and when they should guess and move on quickly. With practice they begin to think like the test writers. They become more confident and less affected by the stress that makes students do poorly.

At present we do not know of any college exam practice programs available for the Macintosh, but in the back of this book we list a number of companies that produce such programs for other computers. You can write or call them to see when and if they will be converting their programs to the Macintosh.

When it comes to practicing for college entrance tests, the computer is a great asset. It is a patient teacher, willing to spend as much time with your students as they can bear in any one sitting. Many of these programs include a clock so students can learn to judge how much time they are taking on each question. In some programs students can ask for immediate feedback on questions that they answered incorrectly, or they can wait until they are finished with the series and then review each of the incorrect answers. A good program will tell the student not only what the correct answer is, but why it is correct. In other words, such programs help the student to think correctly.

When selecting programs to help your students prepare for college entrance exams, you should use all of your personal experience and knowledge about these tests to choose the program that will help them the most. Don't just buy a program based on the promotional material—try it out.

Since these programs can be shared by many students, and will last for a long time, you can afford to pay a little extra if necessary. But don't pay more than you have to. If
there are no programs available for the Macintosh, you might want to make it a class project to write one based upon one of the practice books that are available.

Writing drill and practice programs: This is another subject that we will discuss at length in the section on using the computer to teach programming. Here we just want to say that because drill and practice programs are the easiest variety to write, you can use existing programs to teach students how to write their own.

Warning: Although drill and practice programs are valuable in their proper place, they may be relied on too heavily by teachers. It is hard to justify the cost of computers in the classroom if the only thing they are used for is something that can be done just as well using traditional teaching tools. The fact is, these programs barely touch the capabilities of the computer. They should be purchased and used sparingly and only when there are no other more important demands for computer time. When you start looking for educational software it may seem like these are the only programs available. And it would be easy to make your purchases from this group and call them good. It's probably even easier and less painful to learn how to run these programs. But don’t take the easy way out. Your creativity is essential in planning a computerized classroom and drill and practice programs are only the beginning of what is possible.

TUTORIAL PROGRAMS  Description: The tutorial is an educational program that utilizes written explanations, descriptions, graphic illustrations, questions, exercises, and problems to develop and teach concepts. These programs may come with pretest materials to determine which lesson a student should begin with. The lesson itself resembles a dialogue in which the program presents some information and then asks a series of questions to see if the student has understood the concept. The student answers the questions and then, depending upon his or her response, the computer provides more information and asks more questions. Tutorials give students more flexibility than drill and practice programs, which require an absolute correct answer. However, even these programs can be somewhat limited unless the programmer has spent a good deal of time considering all the possible responses that might be given. Frequently, tutorials are written in a multiple choice format to narrow the choices.

Some educators are critical of tutorials because of this limited capacity to respond to a variety of student answers. They are concerned that important concepts might be trivialized by the limits of the program. However, if tutorials are used as a supplement to other modes of teaching, such as simulation programs (which we will be discussing next), lectures, films, reading, experiments, demonstrations, and class discussions, this should not be a problem.

After the tutorial part of the program has been completed, drill and practice exercises are frequently used to determine the student’s progress. Finally, a posttest for each objective or group of objectives is given to determine the student’s mastery of the subject.

Since tutorial programs must be able to predict a variety of possible correct responses, allow for insignificant misspellings and capitalization errors, (you don’t want a minor problem to stop the progression of the lesson), and respond to incorrect answers with an explanation and additional learning experiences, they require much more complex programming skills than drill and practice programs. This means there are fewer tutorials
available to choose from. It takes a fairly good programmer who knows something about learning theory to write a good tutorial. So be sure you examine this type of program carefully before you make your purchase.

There are several types of tutorials. Some progress in a linear fashion, presenting a series of screen displays to the user, regardless of individual differences. In these programs incorrect responses may call in a reteaching sequence, but the attention paid to the individual user is not as sophisticated as in the second kind, or branching tutorial.

A branching tutorial does not require users to follow the same path. It can direct students to certain lessons or parts of lessons according to results of computerized pretests, or to the answers given to certain questions within the program itself. Generally, the tutorial automatically records student scores so that they are readily available to the teacher.

Even though tutorials provide an “interactive” relationship between the student and the computer and can allow for individual differences between students, they do not use the full capabilities of the technology. Tutorials and drill and practice programs can be a real asset to the classroom, but let’s not stop at that.

**The computer as tutor:** Tutorial programs show up in all kinds of classroom settings. They might be used to teach students concepts of physics, social studies, history, political science, math, or grammar. They can be used to teach foreign languages or computer literacy. They can even be used to teach students and teachers how to operate the computer itself. A programmed tutorial can be used to supplement teacher lectures in almost any subject. This doesn’t necessarily mean there are tutorials available for every subject, or even that the ones available are worth the expense. But if you find a good one, you will have found a valuable assistant.

**Tutorials for Applications Programs:** There are a number of tutorials available that were initially developed to teach professionals how to use their computers and applications software. These tutorials are also valuable for students, particularly in courses that teach about real-world situations. The English class that is producing a newspaper or writing papers, stories, and other assignments can take advantage of tutorials to quickly get up to speed on a word processor. Business students who work with a spreadsheet tutorial can get right down to performing accounting functions, calculating tax returns, and making financial predictions, rather than spending long hours learning how to operate the programs.

Teachers who plan to use applications programs for writing correspondence and recordkeeping will also find these tutorials valuable timesavers. It is probably only necessary to purchase one or two copies of this type of program per classroom. Students can share the disk after you have done an initial run through on a demonstration display monitor in the front of the classroom. If you have a number of MacWrite programs available, students can use those on their individual computers to follow along and try out the concepts being taught on the tutorial.

**Simulations** **Description:** This is where the computer really shows its stuff. Computer simulations allow the student to experience situations which would be impossible to duplicate with other educational media. These programs bring the abstract concepts
of theory into the world of practical application. True, these experiences are "simu-
lated," but where else could a student get the chance to sit behind the flight panel of a
747, or make decisions in the boardroom of a Fortune 500 corporation?

The most important aspect of educational simulations is that they require involve-
ment. Whether the student is learning to drive an automobile with one of the driver educa-
tion programs or conducting scientific experiments in a simulated laboratory, that student
must make decisions, and those decisions effect outcomes. The student must think logic-
ally and consider all the variables of a situation. He or she must develop strategies, and
once these strategies have been implemented face the consequences of those choices.
Simulations go real-life experience one better. They give the student a chance to go back
and learn from mistakes. Decisions can be changed and new strategies developed that
bring a desired result. And the student learns how to think and solve problems in the
process.

A simulated educational program can be as simple as the creation of a financial
statement for a small business or as complex as a flight demonstration developed by
NASA to show certain events that can't be captured by photography. In this area of educa-
tional programming the imagination is the only limit. Microcomputers appearing on the
market these days already have excellent graphic and animation capabilities that make for
great simulations, and it is only a matter of time before we will have highly detailed video
simulations for the Macintosh.

Simulations are a real boon for the math and science teacher. They provide an excel-
lent medium to demonstrate abstract concepts that are often difficult for the student to
visualize. These programs can be used to convert the computer into a science lab in which
experiments can be conducted, data stored, graphs and charts produced, and results ana-
lyzed. Special interfaces can be used that allow the computer to be used with equipment
such as thermometers and other measurement devices allowing the student to analyze in-
put from the real world as part of a simulation.

Simulations can be used to teach driver education and as an aid to the school coach
to demonstrate game strategy. They can be used in a geography classroom to show the
movement of continental plates or the evolution of a volcano or typhoon. In a biology
class they can demonstrate what happens to an ecosystem when certain variables are
introduced.

If you let your imagination go, you can think of a hundred ways that simulation
programs can enrich the educational experience of your students. The truth is, not all of
these programs have been created. In fact, very few of them have been. But the technol-
ogy is there just waiting for educators and programmers to put them together.

EDUCATIONAL GAMES Description: In addition to the three types of programs
listed above, there are educational programs that teach skills and concepts under the guise
of a game. These programs involve problem solving, usually in the area of math or lan-
guage. While many educational games contain the same kind of teaching strategies con-
tained in drill and practice or tutorial programs, they are different in that there are a plot,
characters, and some goal that the student tries to achieve. There is a winner. The user is
involved, but not in the same way as with a simulated program. Although there are conse-
quences for simple decisions, and one answer will lead to a different response from the program than another response, these programs generally do not utilize the complex strategies and multiple variables of simulations. Also, simulations tend toward duplicating real-life situations, whereas educational games generally contain elements of fancy.

Educational games are valuable for developing systematic thinking and skills of association, and for providing a concrete form for abstract ideas. The game aspect keeps the student entertained and makes the learning process painless. Many programs developed for preschoolers use this format.

**Educational games in the elementary classroom:** This program format may be the best one for introducing elementary students and even preschoolers to the computer. Games can capture the student’s imagination and teach while they entertain. They have qualities found in television cartoons, illustrated storybooks, and puzzles. While children are playing educational games, they are also learning how to operate the computer. They find out that when they push a key on the keyboard something happens on the screen.

**Games for older students:** There is some controversy over whether or not older students should be spending their time playing computer games, even if they are educational. Such programs are often associated with arcade games, and parents and teachers are worried about kids spending too much time in arcades anyway. We think that it doesn’t hurt to have a few educational games, and maybe even a few that are not so educational, around for students to use during lunch hour or recess. You might even want to use these programs as a kind of reward for those who get their work completed quickly. The more time students spend with computers, the more comfortable with them students become. At present, the problem seems to be giving all children an opportunity to become familiar with computers, not trying to limit their access. If you have your students share the computer at these times, you can also promote the development of healthy relationships and cooperation.

**Classroom Administration and Tools for Teachers**

The classroom computer is versatile. It will help you with your teaching, and it can also take on some of the tasks that tend to eat up a teacher’s spare time. There are programs that can help you keep student attendance, monitor individual student progress, generate and administer tests, calculate and record grades, make schedules, create programs, and create teaching aids such as maps, calendars, quizzes, puzzles and whatever else you can think of. There are special programs that take you through the whole process of designing and filling out reports on individual education programs (IEP’s) for students with special needs.

**A Testing Machine** What would you think if you could find an assistant who could generate tests for your class (with a different set of questions for each student), administer and correct those tests, record grades, and print out a report for your files? Such an assistant exists in a personal computer. Of course, you’ll have to spend a little time creating a list of test items with one of the testing programs that are available. (If
What Do We Want the Computer to Do?

There isn’t one available for the Macintosh, learn how to write one yourself.) Some of the programs that have been written for other computers allow students to enter comments about the questions on the examination, providing the teacher with valuable information about their response to the material. If time is not a critical element of the test, it is possible for the computer to record the students’ answers for grading, then give immediate feedback as to whether or not the question was answered correctly (and if not, what the correct answer is and why).

Immediate feedback is very valuable for student progress since at the time of the test students are highly motivated and open to learning the correct answer. When children must wait several days to get their tests back from the teacher, this high level of motivation for learning is diminished and children tend to concentrate on their final scores rather than the individual questions that were answered incorrectly. Instead of the test being an individual learning situation, students are made to feel embarrassed or proud of a grade or score.

The reports compiled by the computer on the results of the test can be used to determine what kind of problems the overall class had and also to monitor individual student progress. This information can be used by the teacher to determine if more time needs to be given in class lectures to explain difficult concepts.

If you don’t have a computer for every student—probably the case in most classrooms—you might have a problem giving the test to everyone at the same time in the traditional way. However, if the computer can generate a unique set of questions for each student, this does not pose a great problem. Several small groups of students can take the exam throughout the day. In fact, we can see how this might take some of the pressure off students who freeze up in a test situation when they feel the intense competition of the other kids all around them. The goal then becomes to improve oneself, rather than doing better than the other guy. You might take this one step further by building the testing process into the lesson itself, thus making it a learning experience. This is most practical at this time in testing information learned from a program on the computer itself, since many of them already have built in pre and posttests that can give the student scores after each lesson. As the computer is utilized in more and more of the curriculum—for example, if a teacher gives a lecture, shows a film, or demonstrates an experiment and then turns the kids over to a program designed as a part of the whole lesson package—that program can also become the testing instrument.

A computer program can enhance the inclass lesson by providing review material or by expanding upon the ideas being taught. It can be used to help students see the same ideas at work in a variety of situations or contexts. This act of generalization is important to the process of thinking itself.

**Programs for Teachers**  
**Word Processing:** You’ll have to decide if you need your own copy of a word processing program, or if it will be sufficient to borrow one from the student software library. If you must compose reports, correspondence, lecture handouts, lesson notes or any other written materials, you will find a word processing program an incredible timesaver and organizer. If you’ve ever used one long enough to know what you’re doing, you don’t need to be convinced. Once you get your hands on a word processor you will be much more likely to produce special written materials for your stu-
If you have a computer at home that is compatible with the one you use at school, you'll have the added convenience of being able to take your work home on diskette.

**Authoring languages:** An authoring language is a high-level language, such as PILOT, which allows a teacher to write programs even if they have little knowledge of a lower-level computer language such as BASIC or PASCAL. This means that the teacher can easily create programs to keep track of student records, generate tests, and write drill and practice or tutorial programs that meet the specific needs of the classroom. For a teacher who doesn’t have the time to stop everything and learn BASIC or one of the other programming languages, an authoring language is an invaluable tool. They shouldn't be seen as a replacement for learning programming skills, but only as a tool for certain tasks.

Some authoring languages are built around a questionnaire format. The program asks the teacher specific questions and then, depending on the answers, creates a variety of program formats incorporating the material provided by the teacher. The more sophisticated authoring languages, like PILOT, allow the teacher to add graphics, sound, and special utility routines to their programs.

**Recordkeeping aids:** There are a number of programs that can be used to help you keep track of such things as student attendance, schedules, and grades, and that can assist you in compiling reports from the recorded information. There are also programs designed to help you create quizzes, crossword puzzles, worksheets, and tests.

Test generating programs allow you to put a large number of questions into a database so that the computer can then generate individualized tests for each student. The computer program can then administer the test, record grades, and come up with a report of all grades including class average, high and low scores, and lists of specific questions answered incorrectly. This last information can help you to know if there are certain areas that many of the students are having problems with that you should spend more time on. It also lets you know where certain students are having specific problems so that you can work with them individually, referring them to text materials or having them work with a drill and practice or tutorial program that deals with the subject.

**Teaching Computer Literacy**

What is computer literacy? In an article entitled "Computer Literacy the What, Why, and How," which appears in Dale Peterson's *Intelligent Schoolhouse; Readings on Computers and Learning,* (Reston Publishing Company, 1984) Arthur Luehrmann examines the word literacy as it is used in the context of language and mathematics and arrives at the following definition for computer literacy: "Computer literacy must mean the ability to do something constructive with a computer, and not merely a general awareness of facts one is told about computers. A computer-literate person can read and write a computer program, can select and operate software written by others, and knows from personal experience the possibilities and limitations of the computer."

This is the definition we will be using when we speak of computer literacy. If you are teaching a "computer literacy" course in this sense of the term, you will be teaching students not only how to operate prepackaged programs, but also how to select those pro-
grams and pick the best available, how to write their own programs, and how to debug those programs and make them operable for other users.

We think it is important to take full advantage of the computer's capabilities no matter what age your students are. If your school has a special course in computer programming, teachers from other classes can help students apply their programming skills to other subjects such as mathematics, science, or English. Although we doubt that very many elementary schools will have special programming courses for their students, this does not mean young children can't learn how to program. In fact, if you are using a language such as LOGO (which we will talk more about later) this may be the best time to start teaching programming. Young kids have none of the inhibitions that make it difficult for older children and adults to learn about computers and programming.

Every teacher who has computers in his or her classroom should be able to teach some basic programming skills. Certainly, this will not happen immediately, since teachers have to learn how to program themselves. Still, if computers in the classroom are to become really cost effective and a central tool in the process of education, they must be used to full advantage.

**Learning the Language:** If you have a limited number of computers for your computer literacy course there are a number of things you can teach your students without them having to use computers. For instance, there are a large number of terms which must be learned by the student if they are going to be able to understand and speak the language of computers. Here we are not referring to a programming language, but the language used to describe the various parts and functions of a computer. Until students learn what the words mean, the world of computers will remain a mystery. Although it might be more interesting for students if they could learn computer terms from the computer itself, it can be done using the tradition pen and paper method.

Once students learn the various commands of a programming language they can do much of their writing on paper and then enter their programs into the computer when there is one available. Make sure your students get some time on the computer when they are first learning the commands. It will make it much easier to learn what each command does if they can see it on the screen. While students are learning to program, they will frequently stop to run the program to see how it is going and then continue writing. It is interesting to note that most professional programmers use the paper and pencil approach to programming. They have all the commands down cold and know just what will happen, so it is no problem for them to write away from the machine.

**Teaching Computer History:** Although it isn't absolutely necessary for a student to know how and when computers came on the scene in order to be able to use them, some background on the development of computers is valuable. Giving students a historical perspective provides a context for understanding the impact computers are having on society. It's also valuable to have your students consider what kind of changes will occur in society as we move into the "information age." What exactly does that mean? Are the changes already evident? We have seen many layoffs in the automobile industry, in steel mills and copper mines. Are these things happening because computers and robots are taking over? Or are we just not competitive with other countries, such as Japan?
Should we try to hold off the effects of computer technology? Can we hold them off? If we can’t, how do we go about training people for new jobs? Should the government be involved? What is the responsibility of schools? You can see how discussing computer history can lead to all kinds of interesting and important discussions.

**Programming:** Programming taught in the context of a computer literacy course should focus on making the activity interesting and enjoyable for the students. Here we are not thinking of the highly mathematical approach to programming found in the computer science department of a university, but the kind of programming that someone might learn at home working with a personal computer. Students who take these basic courses and find they have a real aptitude for and interest in programming should be encouraged to take higher level courses where they can develop advanced skills.

The programming language generally used for computer literacy courses is BASIC (Beginners All purpose Sequential Instruction Code). BASIC was originally designed as a language novices could learn quickly and use for a variety of programs. It is less structured and easier to use than some of the higher-level languages such as PASCAL and FORTRAN. And it is available for almost every microcomputer on the market. Apple and Microsoft Corporation both have a version of BASIC available for the Macintosh.

Some people complain that BASIC isn’t what it’s cracked up to be and that it gets unwieldy in longer programs because it is not structured. In response to these complaints, the original developers of BASIC, John Kemeny, former president of Dartmouth College, and Thomas Kurtz, a professor of computer science, have decided to create a new version of BASIC to be called TRUE BASIC. TRUE BASIC is being designed for computers with 128K of RAM or more and is being developed specifically for educational uses. It should be available for some computers sometime in late 1984. We’ll have to wait and see if someone develops it for the Macintosh. The new BASIC will have a built-in screen editor and more sophisticated graphics capabilities which should make it competitive with LOGO’s graphics.

There are a number of other programming languages which we will discuss under the section on Teaching Programming.

**Computer Ethics:** Presently there is legislation before Congress to decide how to deal with a new form of crime that is appearing with the spread of microcomputers into homes, schools and businesses. Has a crime been committed and what should the penalty be if someone using a computer destroys the records of a company, steals information, or invades the privacy of another citizen? These questions must be answered by lawmakers and should be discussed with students who are learning about computers.

Another question of ethics arises when students ask if they can make copies of prepackaged software to take home for use on their personal computers. Should it be allowed? If not, how should a teacher explain this to students (particularly if students have seen their parents making copies of programs at home)?

Since questions of computer ethics and computer crime are so much an issue, students can be encouraged to look in newspapers and magazines to find out what decisions are being made. They should also be encouraged to devise their own solutions to problems and questions that come up as they use computers.
Note: We have described a computer literacy program as if it were taking place for the first time at every grade level. That is, when we talk about providing programming courses, or historical background, we are assuming that students of any age have not studied the subject previously. Once computer literacy becomes a standard subject in primary and secondary schools a curriculum will have to be established that builds on the material learned the previous year.

We are at a difficult stage right now, with everyone having approximately the same level of computer literacy, including teachers. Programs now being developed around computers will have to be constantly examined and restructured to allow for a student population with increasing computer skills.

Teaching Programming

We have already discussed teaching programming as part of a computer literacy course and we have suggested that BASIC is the language most frequently used. Here we want to describe some of the other languages and their specific applications.

WHAT PROGRAMMING LANGUAGES WILL YOU NEED? Although BASIC is the programming language that people seem to know about long before they know anything about computers, and is also the one that gets used in most computer literacy courses, it is certainly not the only one available. In fact, it isn't necessarily the best language for your students.

LOGO: A Programming Language for Kids: If you want to start teaching your elementary students to program, you should be sure that the computers you buy will run LOGO. (Rumor has it that there will soon be a version of LOGO out for the Macintosh.) This programming language was especially designed with children in mind. Though it is a powerful language that even experienced programmers can appreciate, it is simple enough for young children to use.

LOGO was developed by Seymour Papert at the Massachusetts Institute of Technology. It uses a "turtle," either in the form of a turtle-like robot (hence the name) that moves around on the floor or as a "turtle character" on the screen. This turtle teaches the child, on an intuitive level, some of the basics of Euclidean geometry. "The idea of programming is introduced through the metaphor of teaching the Turtle a new word," Papert says in his book Mindstorms (Basic Books, Inc., New York, 1980).

The child learns to program by instructing the turtle to draw lines either on the screen or on a piece of paper placed below the robotic turtle. By typing in a code and a number the child programs the turtle to move a certain number of steps, say FD (forward) 20. To change the direction of the turtle's movement the child can type in a direction such as RT (right) or LT (left) followed by the number of steps the turtle should take. These directions are based upon the idea of 360 degrees in a complete turn; thus, if the child enters RT 90, the turtle draws a right angle. The child can then move it forward a certain number of steps, command another RT 90 turn, and so on until the turtle has drawn a square. Of course, the first few times a child directs the turtle's movement he or she will
not come up with a perfect square or rectangle, but by experimentation and repetition the child will soon learn how to give the necessary instructions to have the turtle draw squares, triangles, circles, and other geometric forms.

Often when kids are using the robotic turtle you will see them moving themselves in space in order to decide what they want the turtle to do. This is a very practical way for them to come to a concrete understanding of what is going on with the turtle. Once children have learned to move the robotic turtle around the floor, they can easily transfer what they have learned to the turtle on the screen. But you don't have to buy the rather expensive robot; they can also learn to move the turtle on the screen from the beginning.

Certainly children can learn to draw squares on a piece of paper, and at a fairly early age, but by moving the turtle a certain number of steps forward or a certain number of degrees to the right or left to draw the figures, they learn about the mathematical nature of geometric forms rather than just how to draw a square. The turtle can make the abstract ideas of Euclidean geometry concrete, and can do it for children who are generally considered too young to learn such abstract ideas. And the best thing about it is—IT'S FUN!

Once children have learned to program the turtle to draw a geometric form, they can name that form and store it in the computer's memory or disk. The program can then be called up and they can see the geometric forms drawn on the screen by running the program. By programming the turtle to draw several forms in combination, students can begin to create pictures. For example, they can write a program instructing the turtle to draw a square with a triangle on top of it, a graphic resembling the traditional first and second graders' drawing of a house. As students learn more commands, they can create increasingly complex graphics. They also learn how to communicate with the computer through a programming language, which will be a strong base for learning other programming languages such as BASIC or PASCAL.

**BASIC: A Programming Language for Every Student:** As you recall from having read the section on computer Literacy, BASIC stands for Beginners All purpose Symbolic Instruction Code. If only because it is readily available for almost every microcomputer on the market, BASIC is a language every student should learn. One of the reasons BASIC is so widely available is because it was one of the few languages that could fit into the limited memory of the first micros. Now that these small computers have a lot more memory, BASIC is getting some competition. Still for the time being it is somewhat of a standard.

BASIC is made up of a number of commands that instruct the computer to do various things. (See the chapter on Macintosh BASIC.) Although BASIC is a powerful language it is not a highly structured language, and people sometimes complain that programs written in BASIC are convoluted and difficult to figure out if you don't happen to be the original programmer. Other people say BASIC allows more creativity than the highly structured languages like PASCAL. You'll have to decide for yourself.

If you are teaching BASIC you should keep in mind that because BASIC must be translated into machine-readable code by a compiler or interpreter, and these translators vary from one computer to the next, programs written on one machine sometimes have to be rewritten before they can be used on another machine. Generally the computer you
purchase for your school classroom will come with a BASIC manual specifically written for your computer; you can use this manual to straighten out the confusion.

Another thing that may be of interest is that there are a large number of programs written in BASIC published in monthly magazines or books that your students can use to learn about programming. By entering the code into a school computer students can see how the various commands are used, they can find out how the BASIC for the computer they are using differs from the BASIC used by the original author, and they can learn to debug the program once it has been entered. Later they can write original programs using the skills they have learned.

PASCAL: A Language for Computer Scientists: PASCAL was developed in the 1960s and named in honor of Blaise Pascal, a seventeenth-century French mathematician whose work influenced computer technology. It is often used as the language of first choice in computer science departments since other languages can more easily be learned once PASCAL is understood. If you have students who are interested in going into computer science, you should be teaching them PASCAL since the College Board has recently decided that the exam for advanced placement in computer science will be given in PASCAL.

Like BASIC, PASCAL is easy to understand and use, but it allows for structured programming and includes different sections for dividing a program into segments, (as does COBOL). It is also more powerful than BASIC, although not as powerful as COBOL. Remember, if you want to teach PASCAL you will have to purchase a computer that it will run on. PASCAL is available for the Macintosh.

COBOL: A Language for Business: COBOL is an acronym for Common Business-Oriented Language. It was designed to meet the special needs of business, particularly for data processing and reporting applications. The most frequent use of this language, then, will be in the business classroom or in computer science courses where students are preparing for careers as programmers.

As a programming language for other than business applications, COBOL has some problems. It requires a large amount of main computer storage, and a program written in COBOL tends to be less concise than programs written in other languages. We expect that COBOL will also be available for the Macintosh in the near future.

FORTRAN: A Language for Scientists and Mathematicians: FORTRAN stands for Formula Translation. This programming language has been a great aid to scientists in the writing of mathematical formulas. It is also very quick and precise in performing mathematical calculations. Since this is a very complex language it will probably be taught mostly to students in advanced courses who intend to pursue careers in computer programming, science, or math. It is obvious that a teacher who takes on a FORTRAN course must have a good understanding of the language.

Other Programming Languages: In addition to BASIC, LOGO, PASCAL, COBOL and FORTRAN, there are a number of other languages that are not so familiar. Since the computer you select must be able to run the programming language you wish to teach, these are probably the ones you will be working with. But keep your eyes and ears open to find out what is coming out that is new and valuable for your students.
Authoring Languages: An authoring language is not really a programming language in the sense that BASIC is, but it does allow the user to write programs. Using a series of prompts or questions, an authoring language allows a you to create the format for a program and then fill in the content and presentation of your subject. This allows teachers to have input into the specific content of a program without having to know a programming language. Although an authoring language might be used by a student to create a program, we would not consider it part of a regular programming class. These languages should be considered part of the teacher’s package rather than as student material.

Vocational Training

Computers can be used in a variety of classrooms to prepare students to enter the job market upon graduating. As we have already discussed, computer skills are going to be required for many of the jobs of the future.

Computers and Business

Office Skills: In the past many high school students have taken courses in shorthand and typing so that they would be prepared for office jobs upon graduation. Although typing skills are still required for most office jobs, typing is now being done on computers with word processing or database programs. As personal computers find their way into the offices of managers and administrators, secretaries may find that, rather than taking dictation by hand or off a dictation machine, they will be revising letters and correspondence that have been written on diskettes by their bosses. The process will be one of editing and revising material on the disk with a word processing program. Even if secretaries must take dictation in the traditional way, they will find that a word processor saves them time and allows them to present error-free copy to their employers.

Since the computer and word processing programs are pretty much replacing the standard typewriter in the business worlds, we think that classrooms where office skills are being taught should be outfitted with computers for every student. It is our belief that all classrooms where office skills are being taught should be outfitted with a computer for every student.

Bookkeeping and Accounting with Computers: With all the database and accounting programs available for computers, high schools can teach a variety of vocational skills using the computer. Since programs such as VisiCalc, Lotus 1-2-3, and dBase II are being used by accountants, office managers, financial analysts, scientists, and engineers to solve problems, students who learn how to use these programs are preparing themselves for future careers. In addition to the vocational skills they will learn, students can be taught such practical skills as balancing their checkbooks, computing taxes, or developing a personal budget. They can have fun keeping track of anything from stamp collections to their daily caloric intake and output in a personal health program.

Sometimes learning how to operate these business programs can be a real chore because of the poor quality of the documentation. Something you might want to keep your
eyes open for are business programs that come packaged with tutorials. They are being developed especially with students in mind. Also, there are a number of tutorials being produced by third party vendors.

**Preparing for a Career in Computers:** Obviously anything that students can learn about computers will aid them in careers as programmers or engineers in the computer world. In addition to teaching programming, a high school might provide a course on computer electronics. Of course for this class you would have to have a real pro teaching. If you did teach such a course, your students would be very valuable in helping to maintain the rest of your school’s computer equipment. Many of these students would become knowledgeable enough about computers from studying in class and outside learning that they would be able to give input into what kind of computers you should purchase if you come up with funds to expand your system. It might not be a bad idea in any case to have students as part of the decision-making panel when it comes to purchasing hardware and software for your schools.

**Communicating With Other Computers**

One of the most exciting things that the computer offers and that makes it an essential tool in the “information age” is its capacity to communicate with other computers either through local networks or telephone lines. Although the technology is developed and being used to some degree, telecommunication in education has barely been tapped at all.

Some people dream of a society in which children and adults alike will be able to stay at home and study any subject they choose with a wealth of resources, great teachers, and the world’s largest libraries at the beck and call of their personal computer. Already there are large online databases such as The Source which have categorized information on any subject from the stock market to current medical research. We will have to wait to find out just how far this technology will take us. Perhaps our children’s formal education will take place at home, and school as we now know it will disappear. Or, maybe teachers will pick up on telecommunications technology and make our schools better than they’ve ever been before.

**A Computer for Special Students**

A computer is an invaluable device for allowing handicapped and learning disabled students to enter the regular classroom. Not only is it an incredibly patient and consistent teacher, but it can provide continual reinforcement and feedback, it does not embarrass the user when mistakes are made, and it has an added advantage of being adaptable to a variety of users with special problems. Computers can be used to operate wheelchairs and can speak to those who are blind; those students who cannot speak can still learn and respond to questions by entering them on a keyboard. For those who cannot manipulate a regular keyboard, special input devices such as joysticks, mice and light pens can be used.
One group of students who can benefit greatly from learning to use computers are those who suffer from dyslexia. Since these students generally have problems drawing all types of symbols, printing letters for written communication becomes a difficult and often impossible task. When a student with dyslexia has finished writing a paper and goes back to correct errors, new errors will often be made, and in correcting those others are made, until the student despairs of achieving an error-free copy.

Working with computers and word processing programs, dyslexics have an opportunity, perhaps for the first time, to write down their thoughts and make the necessary corrections without having to type or write the whole thing over again and take the chance of making new errors. Word processors separate the act of writing from the difficult act of printing. In addition to the delete and insert functions of word processors, which allow the dyslexic to clean up mistakes, the fact that words, lines, and paragraphs can be easily manipulated and moved around on the page allows for a new sense of control over organization and development of ideas. Since dyslexics can also have problems in these areas, a word processor can help in the overall organization and presentation of written communication. Selecting a word processing program with clear and concise instructions will help the student pick up keyboard skills quickly so he or she can experience a new sense of power and confidence.

Although handicapped students must generally wait until a technology is developed and perfected for use by the general public before it is adapted to their special needs, it is interesting to note that the modem was originally invented to allow the deaf a way to communicate over telephone lines. While very little of what is possible in the way of educational tools for the handicapped has been developed, research is being done in many areas. Some day all students, no matter what their special needs, may have an opportunity to learn and develop their full potential. Computer technology will be a large part of the fulfillment of that dream.

WHAT HARDWARE AND SOFTWARE WILL WE NEED TO MEET OUR TEACHING OBJECTIVES?

Get the Most Computer for Your Money

This is an overriding objective for most schools since computer funds are hard to come by. Before you start purchasing your computers be sure you know how much money you have in your computer budget. Remember that in addition to buying the computer “hardware,” including printers, modems, and a variety of input devices, you must save money for a variety of programs that are appropriate for your classroom. Some programs are fairly expensive, so be sure you keep these extra expenses in mind when you are planning your budget. There are also ongoing expenses for things such as printer ribbons, paper, diskettes, and maintenance.
How Many Computers Will You Need?

The answer to this question depends on what you plan to use the computer for, but we do have a few general guidelines to consider for all situations.

Computers and Young Children

Since young children have short attention spans and we want to be sure we introduce them to the computer in a positive way from the very beginning, the ideal situation would be to have one computer for each child. Although it is not necessary to have huge amounts of memory—128K is enough—and it is not critical that you have disk drives, you should have color monitors and lots of colorful software with animation and sound. Though at present the Macintosh does not have color, the high-quality graphics makes up for that somewhat.

If you are teaching young students about computers the Macintosh mouse will help them gain immediate access to the computer without having to learn the keyboard. You might also want to check into the availability of other input devices such as graphics tablets, light pens, and joysticks (see input devices later in this section.) Even if they have a variety of input devices available, it is important for children to learn to use the standard keyboard as soon as they are able. Since the keyboard is the input device they will most frequently be working with in occupational settings, they should learn how to use it before bad habits set in.

Computers in the Business Classroom

As we mentioned in the previous section, we think it is important to have a computer for every student in a business classroom. Traditionally in such classes there has been a typewriter for every student, so it should be no different when computers are introduced. The computers for the business classroom should have at least 128K of memory and two diskette drives. You should also have several printers in the room so that you can get hard copy of student work. We would suggest that you get one of the more powerful word processing programs since MacWrite is most effective for short pieces.

Other Classrooms

In biology, English, math, or social studies classrooms you should have at least one computer for every four students. Install them at the back of the room or someplace where students can work on programs or assignments while the rest of the class is engaged in other activities.
If you don’t need computers in the classroom every day, they can be put on carts and moved from one classroom to the next as needed. In each of these classrooms there should be at least one printer. The computers should have at least 128K of memory so that they can run some of the more sophisticated programs.

For other classroom settings get a count of the number of students who will be using the computer and try to determine how much time they will need to accomplish the tasks you assign them. You should keep in mind that when computers are initially introduced it takes a certain amount of time to get up to speed. You should calculate the number of computers you will need based on a smooth-running classroom in which the students have learned how to operate the computers.

**Will You Need a Printer?**

A printer becomes an essential part of the classroom computer package if your students are going to be using computers to do word processing and graphics programming or if you want to use it for recordkeeping and test generating. A dot-matrix printer is sufficient for most school applications. In order to use the computer in a variety of ways, you will probably want to have at least one printer readily available for each classroom. If you are going to be using computers in a English classroom where students will be generating a large amount of written work you may need several printers, preferably ones that have a variety of programmable fonts.

**How Much Memory Will You Need?**

The 128K of memory available with the standard Macintosh is sufficient for most educational programs; however, it won’t be long before every microcomputer will have at least 256K. As educational programs get more and more sophisticated, they will increasingly demand more and more memory. Right now you can get a Macintosh or Lisa with 512K of memory. When you are deciding how much memory you need for your classroom computers, think about the kind of software you will be using. Find out from the manufacturers how much memory is needed for that software and make your decision based upon that information. Remember, you can usually add memory later on if you find you need it.

**What About Diskette Drives?**

A computer with two diskette drives allows you to have your program loaded in one disk, while you create files and write on the other one. When there is only one drive, you must load your program, then put in a working disk, and sometimes stop and put in the program disk while you are working. If you have the funds to purchase the extra drive we would suggest you do it, particularly if your students will be using some of the powerful database or word processing programs that require a lot of memory. If you don’t have the money for the additional diskette drive, don’t worry; one will work just fine once the students get used to switching diskettes.
Portability

The Macintosh is a great computer if you're going to be sharing computers between several classrooms. It is very light and transportable. If teachers want to borrow the computer to do some work at home, it is a very convenient computer to have.

Input Devices

Joysticks, mice, light pens, and graphics tablets are all devices which allow you to give input to the computer without using the keyboard.

A joystick resembles the input device used on arcade videogames. Using buttons and a lever the user can move the cursor quickly around the screen, shooting down space invaders or running away from the chomping jaws of a monster. Generally, with personal computers joysticks are used for the same purpose, that is, for programs written in a game format. Since many educational games for young children use the game format, you might want to have a few joysticks on hand that can be used for this application. Also, there might be some good simulation programs that would benefit from the use of a joystick. If a joystick is required for a particular program, the documentation will let you know.

If you've purchased a Macintosh you are very aware of what a mouse is. If you've never used another computer, you don't know that the mouse is a relative newcomer to the computer scene. It is very much a part of the Apple design plan to make the Macintosh easy to use.

A mouse can be a valuable input device in a classroom. Students can quickly learn to work their computer if they don't have to worry about typing on the keyboard. It helps to sustain interest in young children and is particularly helpful for handicapped students who can't manipulate the standard keyboard.

A light pen allows you to input data by touching a particular area of the computer screen. This device is another helpful tool for handicapped students. It allows even finer movements than the mouse or joystick.

A graphics tablet is a board with a plastic surface that hooks up to your computer and allows the user to create drawings, play games, and learn a variety of subjects from music to social studies. Whatever the user draws on the tablet appears on the screen. Thus it becomes a valuable input device for students who cannot operate a keyboard and also for young artists who wish to do some freehand drawing. Some of these tablets have touch-sensitive surfaces which will respond to the touch of your finger, while others require an electronic "stylus." Tablets with mylar overlays allow you to put drawings or photos underneath the overlay, trace them and store them on disk or print them out. Some of the tablets have very high degrees of precision and accuracy and are used by artists, engineers, and architects. In choosing a graphics board, you must determine whether or not it will operate on your computer.

You can use a graphics table to teach spacial relationships to young children or students with learning disabilities. Drawing on the tablet and seeing what appears on the screen are good for teaching eye-hand coordination. And the teacher can work with stu-
students on following instructions by requesting that each student draw a specific item on individual tablets. Of course the children will have a ball just drawing their own pictures and filling them in with the many patterns offered by MacPaint.

Graphics tablets can be used in geography to draw maps, or in geometry to study geometric shapes and proportional figures. Young architects can use the tablets for drafting sophisticated designs and schematic drawings. Since this is a relatively new technology, the full potential for educational applications has not been developed. Again, as we have said so often, your imagination is the only limit to what you can do with this computer technology.

What Programming Languages Will You Need?

We have already described in some detail the various programming languages you might want to choose for your students. Before long the Macintosh will have many programming languages to choose from. You will probably want to have more than one language for your school computers. It is particularly important to have BASIC, PASCAL, and LOGO if you are teaching young children.

Communicating With Other Computers

If you want to enter the world of telecommunications and allow your students to access online databases you will need the equipment listed below. The first thing you should keep in mind is that your telephone bill will go up quite a bit. This is the highest ongoing cost in operating a modem. In fact, the cost might be prohibitive for many classrooms unless a WATS line or discount telephone service is available. Some of the online databases have toll-free numbers, which should alleviate some of the problem.

To get your classroom online you will need a microcomputer with at least one diskette drive to store the information you request from the various databases.

You will also need a modem, the device that allows communication to take place between one computer and another. It hooks up to your computer and translates the signals coming across the phone lines to a language your computer can understand. (Refer to the chapter on modems for more specific information.)

It is not enough to have a modem; you must also have software that allows your computer to work with the modem. The software you purchase will depend on what kind of computer you want it to run on. Some software will allow your computer to act only as a terminal. That is, you can type messages back and forth with the computer you are communicating with, but nothing will be stored in your computer's memory. Other programs allow your computer to capture or store data that can be transferred to a diskette or printer.

In addition to connecting you to online databases such as The Source and CompuServe, a modem will allow you to communicate with computers in other schools or
libraries in your area. You can set up an electronic mail system between schools or school districts.

It is also possible that in the future the school computer will be hooked to the student's home computers via a modem. In fact, this is already happening in the Mountain View–Los Altos Union High School District in California. (Classroom Computer Learning, March 1984). Using modem-connected computers, parents can track student progress from their own homes. Students can gain access to a school calendar. At present the system is being used minimally, but you can easily imagine the possibilities of a connection between home and school. Another happy thought: if students and parents are only calling across the city, the cost of phone usage will not put such a system out of the realm of possibility.

If you have one of your computers hooked up to a modem, you should also have it hooked up to a printer so that you can get hard copy of information you want to make available to all your students. A dot-matrix printer is sufficient for this task.

**A Computer for “Real-World” Applications Programs**

There are a number of applications programs on the market that are valuable teaching tools for the classroom. By teaching your students to use word processing or database programs you are giving them skills which will help to support them as they go on to college or enter the job market. Most of these applications programs require powerful equipment with at least 128KB of memory. You will also need to have a printer available. These applications programs are fairly expensive, but since they will be used for many years by many students they are a good investment.

**WHAT WILL IT COST TO PURCHASE COMPUTER EQUIPMENT?**

The answer to this complicated question is dependent upon the answer to the previous question of what you want the computer to do. Here we will discuss some of the hardware and software requirements for various applications as well as other things you should consider when working out your budget. We will not try to give you specific costs for each component of your system since prices vary so much it would be hard to give you up-to-date figures. We will, however, give you a list of things to budget. You can fill in the exact prices by checking with dealers in your area. Remember you can often get a lower price if you purchase a package deal.

Though you should purchase some software with the initial hardware purchase so that you can put the computers to work when they arrive in the classroom, you might want to wait until you have more knowledge about what a “good” educational program is before you spend a lot of money. At the very least, you should look to some of the educational-software review publications listed in Appendix D, so that you aren’t making
your decisions blind. We provide a partial list of the software that will run on the Macin­
tosh in Appendix D, but things change quickly in the computer industry there will be
many more programs to choose from by the time this book is published. You should not
take the list we provide as a recommendation, but only as a listing of what is available.

Once a price list has been developed for the hardware, peripheral, and software
components, it should be distributed to all teachers planning to introduce computers into
their classrooms.

Each teacher should create a budget for his or her individual classroom. When these
have been completed, an overall budget can be prepared for the entire school. Give prior­
ity in each classroom budget to those applications that will best serve most of the students.
Some of your school computer needs may have to wait until additional funding can be
located.

It might be wise to have teachers do an ideal budget for a computerized classroom
and also a more economical version. If you justify your computer needs by individual
classroom it may be easier to obtain funding from a variety of sources. You can match up
your needs with your donors. For example, you can ask businesses or parents to help
purchase computers for vocational courses, or you can write a grant to get computers for
special education needs.

When designing your computer budget for each classroom setting you should deter­
mine which of the following components you'll need and how many of each is necessary.

______________________________
Hardware Components:

_____ 128KB RAM Memory
_____ 512KB RAM Memory
_____ One Diskette Drive
_____ Two Diskette Drives
_____ Modems
_____ Joysticks
_____ Light Pens
_____ Mice
_____ Graphics tablets
_____ Dot-Matrix Printers
_____ Letter-Quality Printers

Interface Cards and Other System Requirements:

_____ Printer Card
_____ Input Device Card
_____ Modem Interface Card
_____ Cables for Peripherals

Software Considerations:

_____ Drill and Practice Programs
_____ Tutorials
_____ Simulations
Many peripherals require the purchase of an interface card and special software. You should be certain when you budget for these items that you include the cost of the interface as well as special cords and other materials. Some computers come with peripheral interfaces built-in. Ask the salesperson about these hidden costs.

WHERE CAN WE FIND THE MONEY TO BUY COMPUTERS FOR OUR SCHOOL?

Finding the money to purchase computers for your classroom is not an easy task. Even though many people think it’s a good idea for students to receive training on computers, and feel that it is the school’s responsibility to provide this training, there isn’t a lot of readily available money around to support that idea. But if you look hard and consider all the possible sources, you should be able to get the money together for at least some computers, and you can purchase more later when you have other funds.

Funding from Local School Districts

We can’t say too much about funding from this source, since school boards vary greatly from one area to the next. However, if you examine the kinds of programs your district has funded in the past—perhaps programs for remedial or handicapped students or curriculum-enrichment programs—and then build a proposal for computers around these areas, you may be able to obtain funds from this source. It is important for you to put together accurate cost projections and a schedule for longterm and shortterm plans so that decisionmakers are confident that you can make the program successful. Another thing you’ll want to do is get plenty of support from other teachers, administrators, parents, and community members. And, last but not least, build excitement into the proposal and your presentation of it. It’s easy for people to be bored about computers, particularly if they know nothing about them. It’s up to you to help them envision the possibilities.
YOUR PTA IS OFTEN WILLING AND ABLE TO HELP  If the parents in your community are interested in getting computers in the schools for their kids, there is no reason why you shouldn’t ask them to help make it possible. In more and more states PTA chapters are getting involved to purchase computers for their schools. You might want to get parents involved in planning some kind of meeting or event (perhaps a local computer store will do a demonstration) to get everyone excited. One thing we’ve noticed is that people don’t get excited about computers until they know something about computers. Try to come up with a situation in which people can get their hands on computers or watch their children excitedly work with an educational program. You can combine a fundraising event with an educational event.

Getting Local Businesses Involved

Local businesses are highly motivated to assist schools in purchasing computers since they are dependent on the labor force that comes out of these schools. As we mentioned earlier, schools are likely to get pressure from business and industry to train kids on computers. It is reasonable, then, to expect that these businesses should help pay for the equipment. There is another motivations for businesses to help schools, and that is for the publicity and increased customer loyalty they receive as a result of their contributions. You might remind them that these contributions are tax-deductible.

Computer Manufacturers Have Special Reasons to Help

The more people learn how to use computers, the more people will be buying computers for home use. Keep this in mind when you talk to computer manufacturers in your area to ask that they donate machines to your school. Apple has worked out discount programs that help schools to purchase its computers. Find out if you can get in on one of these programs.

Don’t Forget to Talk to Community Leaders and Politicians

Community leaders, service organizations, and clubs can often be relied on to come up with funds if they are convinced it is for a worthy cause. They can organize fundraising events, gathering large groups of people together to work towards the common good of our children’s education. Remember that politicians have kids, too, and if they can be shown the value of having computers in the school, they can get involved in bringing about the necessary legislation for computer funding.
How to Go About Getting
a Grant From Private Foundations

By writing to The Foundations Center, 888 7th Avenue, New York, New York 10106, you can purchase The Foundations Directory and the National Data Book, which list the more than 22,000 private foundations in the United States. Or you can go to your local library and see if it has copies on hand. In the larger cities, the main library will often have a whole section on grant writing with all the necessary publications. The Foundations Director and the National Data Book not only tell you the names and addresses of the various foundations, but also describe the kinds of projects they are interested in funding and what they have funded in the past. Once you get a list of foundations interested in educational computing projects, put together a proposal and submit it to each of them. (Remember, you can build your proposal around another area of interest and just make the computer one component of the project.) There is a knack to writing grant proposals, so you might want to find someone in your school or community who has done it and who can give you guidelines and assistance. Many college and university libraries have staff specialists who can help you with your grant proposals.

Grants From Your State Board of Education

Local community educators may be able to obtain grants for specific educational projects from their state boards of education. It is advisable to check with your school board to find out what grants are available and to inform them of your intention to make a request for a grant, since they must approve your request and like to monitor the distribution of these funds. Though it varies from one community to the next, the grants awarded are generally related to remediation programs, programs for improving the quality of education for handicapped children, and curriculum-enrichment programs. It may well be that your state officials have become interested in promoting computer literacy and would welcome a proposal to incorporate computers into the school on almost any level.

Federal Grants

Funds are available from many government agencies for regular educational programs and for special projects. Go to your library and find a copy of the Federal Register (published daily by the Federal government), which lists all kinds of information about current funding sources and regulations promulgated by the Department of Education. By reading through this publication you can find out what kinds of projects have been funded in the past; then you can use that information to guide you in your own proposals.

There may be other sources of funding for computers in the schools. Look for books and magazines on the subject. (Magazines such as Electronic Learning and Computer Classroom Learning will keep you up to date on various funding sources.)
WHERE WILL WE PUT THE COMPUTERS?

After you have decided which computers to buy, have located funding, and have put in your order, you had better know where you're going to put the computers when they arrive. By now you should have a pretty good idea of how many computers you will be getting and how you are going to use them. It wouldn't hurt to actually do a drawing of each classroom. Take into consideration noise factors—a printer can disturb an entire classroom when it is operating. Also, if you are trying to work with the majority of the students while a few others are using computers, the class may become distracted and watch what the computer users are doing rather than concentrating on the lesson at hand.

You may only be able to purchase a small number of computers that everyone in the school will have to share. If this is the case, you will have to find a centrally located space where you can set up a computer lab. A room off the library (or the library itself) may be the most appropriate location for such a lab. Problems of scheduling will have to be solved so that the computers are put to the most efficient use and to insure equal time for all students interested in learning about computers. If the lab is part of the library, the librarian or other personnel may be able to supervise the use of computers. However, it is more likely that you will need to hire someone to do the supervising or get volunteers (perhaps students) to monitor computer use, check out software, and solve problems. Since it is likely that there will be a shortage of computer time, you might want to consider opening a lab after school hours so that students can do assignments they didn't complete in class or work on individual projects. This lab might also be used for teaching continuing education courses for parents, teachers, and dropout students. You will probably be able to get a volunteer or students to man the lab. Of course, you will have to consider problems of security and the potential misuse of computer facilities.

WHO WILL TEACH THE TEACHERS?

Teachers need training to use this new educational tool to full advantage. Even though the Macintosh is much easier to use than other personal computers, if teachers are to use all of its many capabilities they have much to learn. We are particularly concerned that teachers remember that the Macintosh is a high-powered computer. It would be easy for teachers to just show their students how to use the mouse and a variety of prepackaged programs and forget about the rest of it. If such an approach were taken, students would graduate from school only knowing how to push buttons and operate the software. They would know nothing about the operation of the computer, how to program it, or how to use it creatively.

Many teachers are terrified when they think about having to teach students computing skills. Of course they are terrified. Here is a technology they know nothing about, and suddenly they are expected to use it. It is going to take some time before teachers are totally comfortable with computers. It's going to take time and education.
Education departments at universities and colleges around the country are beginning to teach computer courses. Teachers should take advantage of these courses and attend as many as they can. But that may not be enough. In many cases the teachers of teachers are learning about computers at the same time and may be only a few steps ahead of their more computer-literate students. Getting teachers up to speed is a serious problem and it will take some innovative programs to solve that problem.

Though it may not be financially feasible for all teachers, we suggest that if you are going to be using computers in your classroom, you should get a computer for your home. For teachers who can’t make the purchase or who choose not to, portable loaner computers should be available. The only way to become comfortable with a computer is to spend time with one in a comfortable setting where you can move along at your own speed, experiment, make mistakes, feel foolish, feel proud, and find out what the machine has to offer.

Once you’ve overcome that initial fear, you might want to purchase some magazines about educational computing. Your school should have subscriptions to a variety of these publications. (We list them in Appendix B.) Buy some books on the subject. Ask the school principal about setting up an after-school workshop. Talk to other teachers. Learn together. Share your ideas. The whole process of learning how to teach with computers can be an exciting and stimulating adventure.

Finally, since teachers will be learning computing skills along with their students, there will need to be an adjustment in the way the classroom is structured. Generally, the teacher is the authority on a subject and the student is there to learn what the teacher knows. In this case the student may know more about the subject than the teacher. This being the case, there needs to be an open and honest dialogue between the teacher and the student, whereby the learning can take place in both directions. Why not let the student be the problem solver sometimes? We are not suggesting that students take over the classroom, but rather that they be allowed to demonstrate what they have learned about computers. If questions or problems arise, let each member of the class work on the solution. The teacher doesn’t have to be the one with the answer. Give yourself a break and have fun with it.

HOW DO WE GO ABOUT SELECTING THE BEST EDUCATIONAL SOFTWARE?

The most important component of any computer system is the software. If there is very little software available for the computer you have selected, or if you select programs that are poorly written by someone with little knowledge of learning theory or instructional design, you will find yourself frustrated with the apparent limits of the computer. Because software is so important it should be the primary consideration in the selection of the computer itself.

Here are some things you should consider when purchasing software for your computerized classroom.
Is Documentation Available and Readable?

Documentation refers to the written material that accompanies the program. Look at it carefully. Frequently the documentation will be so poorly written that you won’t be able to get past it to actually run the program. There is a new awareness of this problem in the industry, and more care is being taken in preparing documentation. However, it is still something you should watch out for. The documentation should include the following:

**Hardware Requirements** The documentation should tell you what kind of hardware is required to run the program. This will include such things as the kind of computer and diskette operating system needed, how much memory is necessary, whether you will need a diskette drive or color monitor, and all other hardware and peripheral specifications that are required or will enhance the use of the program.

**User Age and Teaching Objectives** In addition to the hardware specifications, good documentation will tell you such things as what age group the program is designed for and what the teaching objectives are. The age specification should provide a range of users so that you can determine whether the program can be used in more than one classroom setting. The teaching objectives should give you a good idea of what the program is designed to teach, as well as a summary of the approach that is used. Documentation should also let you know how many students can use the program at one time. This information will help you determine if a program is appropriate only for individual use or if it can be used with small groups or the entire class.

**Program Instructions** Sometimes the program itself contains the instructions for its use. However, in most cases your students must rely on the documentation to provide step-by-step directions. This is where the documentation can get really confusing; you should probably sit down in front of the computer before you leave the store to see if there is any resemblance between what is on the screen and what the instructions are telling you.

The instructions should refer specifically to what appears on the screen while the program is running. The use of actual illustrations of the program is particularly beneficial to keep your students on track. Once you have purchased a few programs and have been through the documentation, you’ll get a better feel for whether or not it is any good. But the real test will always be whether reading it allows you to run the program.

All aspects of the program should be explained in the documentation, including explanations of lessons, help menus, and special commands. It is helpful if the documentation tells the student what to expect and describes potential problems and their solutions. A reference card should be provided so that once students have been through the documentation they can proceed with just a quick look at the card next to the computer.
Is the Program Appropriate for Your Students?

There are a number of variables that make the software requirements of one classroom very different from another, even though the same subject is being taught to the same age group. Each teacher has his or her own philosophy and style of teaching. When selecting software, teachers should keep these individual preferences in mind and examine products according to their own individual requirements.

Is the Suggested User Age Accurate?

Is the program appropriate for the specified age and ability group as listed on the documentation? Just because the documentation suggests a certain user age range doesn’t necessarily mean it is right for your students. A program should be difficult enough that students have to stretch some, but not so difficult that they lose interest. Also, in order for a program to be valuable for more than few weeks, it should have various levels of difficulty so that the student can progress.

Does the Material Presented by the Program Fit in with the Rest of Your Curriculum?

Using a computer in the school classroom is most beneficial when the material learned with the computer fits in with what is taught in the rest of the curriculum. Drill and practice programs can be used in the same way workbooks have been used, to perfect skills in mathematics, spelling, and vocabulary. Tutorials can be used as a review for concepts learned from reading or lectures. And simulations can give the student an opportunity to interact with an idea in a real-world way. Even though each of these types of programs can play an important role in the educational process, some programs in each of these categories will supplement your particular teaching style and philosophy better than others. Choose programs you feel comfortable with, and that fit into your overall plan for your class.

What Values Are Being Taught?

A program may be excellent as far as the documentation, the educational content, and the presentation are concerned, but a real bomb when it comes to the values being taught. Ironically, programs that have been designed to operate with new and exciting technologies often enforce old stereotypes. Look for racist and sexist elements in educational programs and make sure you steer clear of them. Also, many programs use elements of violence to make the presentation more exciting. This is especially true of arcade games, and educational games frequently incorporate the same motivators. You will have
to decide how you feel about such presentations. Kids have always played some form of cowboys and Indians or cops and robbers, but this doesn’t necessarily mean that such games are promoting healthy values.

**Is the Content of the Program Accurate and Is Material Presented in a Format Most Conducive to Learning?**

Since much of the educational software presently on the market was not written by educators, you will have to pay particular attention to educational content. Try to figure out what assumptions the programmer makes about learning and whether they fit with your own knowledge of educational theory and instructional design.

**Make Sure the Program Practices What It Preaches**

Make sure the content of the programs are accurate and relevant and that they utilize effective educational methods. Surprisingly, there are programs on the market that, although intended to teach spelling, contain misspelled words. Also, make sure the grammar is correct. The people who write programs are, as you would expect, programmers, and are not necessarily the most qualified people to teach your students English or math. Some publishers now have educators as well as programmers involved in the process of creating educational software, and we should see increasing numbers of high-quality programs on the market. When software reviewers such as the Minnesota Educational Computer Consortium (MECC) rate programs, they give detailed descriptions of what is good or bad about each one. Publishers can use this information to improve their software.

**Are Graphics and Sound Elements Relevant?**

Do the graphics and sound elements of the program supplement what is being taught. Some programs, particularly educational games, use graphics, sound, and animation to keep students entertained while they are learning. We have no problem with this, but if the special capabilities of the computer can be used to present material on several levels simultaneously, why not take advantage of this capability?

**Good Examples Can Make a World of Difference**

When examples are given in the program, are they appropriate to the subject being taught or do they seem to be coming off the wall? If several examples are provided that approach the problem or concept from a variety of perspectives, students can learn to generalize from the one situation to another. In this way students are being taught not only specific skills or facts, but they also learn how to think.
Look for Positive Reinforcement

It is not necessary to criticize the student if a mistake is made in answering a question. One of the great things about computers is that they can allow a child who is having difficulty in a specific area to work independently and avoid the cruel judgments of other children and impatient teachers. If the computer program itself makes the child feel like a "dummy," this advantage is being missed. Look for programs that use positive reinforcement. This doesn't mean if the child answers a problem incorrectly he or she should get a pat on the back. In fact, if the response to an incorrect answer is too interesting, say, with bells and flashy graphics, the child is liable to keep answering incorrectly just to get the response.

Feedback Should Be Immediate

A program should be designed so that the computer does what it does best; that is, sift through a lot of information quickly and come up with a response. The program should be able to look at what the student did and then branch to another part of the program that will display some useful information. We call this kind of process "feedback." Feedback can be either visual or auditory or both, but it should be immediate and give the student a clear idea of whether or not the answer given was accurate.

Is There Flexibility?

Is the program flexible? That is, does it allow growth? It is obvious that if a spelling program has a limited number of words that can't be added to or changed, it will come to the end of its useful life very quickly. Of course, if you are using it in a classroom situation, it can be used by all your students and so has a longer life. But since they are available, why not purchase a spelling program that contains an authoring system that will allow expansion of the original program? That way, programs can be modified to meet the needs of individual students.

What's on the Screen?

Here are some things to watch out for when looking at how the program is presented on the screen.

Does vital information stay on the screen long enough for you to comprehend?
Once you have been through the introductory material, such as the program title, the name of the publisher, and initial instructions, is there a way to get past these quickly or bypass them all together in subsequent sessions?
Are there "white spaces" on the screen so that the text can be easily read? If there is too much text it makes the task of learning seem formidable.
Does the program let you answer questions simply, such as allowing you to type "Y" and "N" for yes or no? Unless the user is learning to type there is no reason to make the simple act of answering a question difficult.
Is there consistency in the way you interact with the computer in a program? For instance, is there a consistent method for entering responses?

Is the program presented in small units that can be easily assimilated? If the message is to be presented in text, a good program will break the presentation into meaningful units. By meaningful, we mean blocks of information that fit logically together, that develop a subject by steps. The key is to find programs that have broken down the presentation into units that can be easily presented in one frame of the computer screen.

Does the program contain help menus that are easily accessible? Sometimes during the process of working with a program a student might forget the instructions and have to start the program over if the menus can't be called up mid-program. Menus should also provide a way for a student to bypass certain operations if they are no longer necessary. Programs written for the Macintosh are likely to be especially strong in helpful and flexible menus.

Some Things to Avoid

You won't always find programs that meet all of your requirements. In fact you will find very few that do. However, there are some things you shouldn’t compromise on.

**Watch Out for Bugs** The only way to find out if a program has "bugs," is to run it all the way through. If you reach a point in the program where the system shuts down, or "crashes," leaving you with error messages, then you have come across a bug. It is a good idea to have the salesperson run a program for you before you purchase it, or at least make sure there are provisions for returning the disk if it does not run. Sometimes a program will "crash" because you have given an unconventional answer, something that the computer program is not designed to recognize. If this is the case, you are dealing with a very "unfriendly" program and should steer clear of it, since it will be a frustration for students as well as yourself.

A program that gives error messages in the form of English sentences such as "Press ESCAPE and continue," rather than a numbered error code, provides a way for the student to continue a program rather than becoming frustrated and wondering what to do next. You will find that even though this type of help is obviously critical in situations where students will be working independently, many programs do not offer such assistance.

**Avoid Electronic Page-Turners** Avoid programs that are little more than expensive page-turners. These are often based on existing textbooks and were not built to take advantage of the computer's easy capabilities. The best of them may find innovative ways to present text material, but sooner or later even the flashy or gimmicky presentations will grow tedious. Kids have little tolerance for these repetitive types of programs. The drill and practice programs may have some problems in this area. But really, they are more like flash cards and there is some usefulness for them when it comes to tasks of memorization. The "electronic page-turners" are programs that present text from one screen to the next; the student could just as well be reading a book.

**Don't Get Caught in an Endless Loop** Avoid programs that lock the learner into endless loops that ask for a response and then give the same message over and over
until the learner guesses the right answer. An error indicates that some or all of the lesson is not being understood, and a well-built program will “branch upon error,” that is, it will go to a special part of the program that will give the learner more information. A branching tutorial than can accommodate the differences between individual users will take care of this problem.
As we have pointed out several times in this book, the key to any new computer’s success lies in how much software becomes available for it: Without the specialized computer programs that allow users to carry out their special tasks, the computer becomes little more than a high-tech paperweight. This fate has befallen more than a few new computers whose much-ballyhooed introductions promised unlimited potential—as soon as the software became available.

The trouble is that no computer has much software available for it when it is first introduced. And, despite the grand claims of the manufacturer, you can’t really be sure of how much software will appear until it actually appears. It is no easy task to develop a useful new set of computer programs, especially for a computer like the Macintosh. The Mac strikes out on a new design path, with its own unique hardware/software methodology which requires new software development techniques. And that requires quite a large investment of time and resources. It also requires some difficult decisions to be made. It would help to be able to predict the future. Will the new computer really capture a large share of the computing market and therefore justify the development costs? Will the new computer attract diverse enough audiences to justify the development of specialized programs? And, circularly, the answer to these questions depends to a great deal on what other software developers decide to do. After all, a new computer won’t capture a large share of the market and appeal to diverse audiences unless there is at least some software already in existence; without at least a minimal base of existing software, a computer can’t even begin its useful life of service.
SOFTWARE DEVELOPMENT STRATEGY

Apple's software development strategy seems to be comprised of two important steps. First, since software developers are more likely to develop new software for a popular computer, it is up to the computer manufacturer, Apple, to successfully market the computer. No problem there: Apple has pushed the Macintosh with a flashy and costly national ad campaign that has probably caught the attention of most everyone who is interested in buying a computer of any sort. And by carefully positioning it as a more productive (and more expensive) computer than their still popular, lower cost, Apple II series computers (but less capable and costly than the Lisa computer), they have made potential customers of the previous buyers of their other computers. As a result, the Mac is selling faster than any comparable computer in history. Through the middle of its first year on the market, Apple was producing the Macs at a rate of 40,000 per month in their new, highly-automated Mac factory, yet the computer was still hard to get. By the end of 1984, they expected to be producing 80,000 per month.

With sales at those levels, the second part of the software development strategy—working hand-in-hand with software developers—can proceed successfully. Even before the introduction of the Mac, Apple had developed a base of unique and interesting computer programs. They did this by delivering a few prototype Macintoshes to selected software developers, and by working with these companies Apple managed to have a fairly useful set of programs available at the time of the computer's premier. Since then Apple has continued to encourage software development by working closely with software vendors. They have instituted a special team to work with any software developer that wants to take advantage of the unique characteristics of the Macintosh and its complicated operating system. In this way Apple can both encourage the development of new programs and guide the incorporation of Macintosh-unique features. Apple not only hopes to build a large and useful library of programs, but they hope also to tie these programs to the windowing capabilities of the Macintosh operating system. In this way, a user can begin using any new program right away, without having to relearn new ways of interacting with the program. In this chapter, we reproduce the long list of developers that have taken advantage of the Mac software development program. We also report some of the directions in which the Macintosh—and its working partners, the software developers—seems to be heading. This future looks promising, although the development of educational programs is, as usual, lagging behind the development of business-oriented programs.

THE FUTURE

Writing a book about a new computer like the Macintosh is not as easy as writing about an established computer such as the IBM PC. The problem is that despite the interest in a new computer, much of what is interesting about a consumer product is yet to happen. It is the history of a new computer—how it is introduced, how the public reacts to it, where it finds its place in the market—that creates the image and the eventual value and utility of the product. Authors of books about new products are forced into the position of predic-
tion and speculation. However, we have been lucky to have a partner in our descriptions and, to some degree, in our speculations: The Apple computer corporation has been, from the beginning, very helpful to us. They have provided us with a computer, advice, and direction, and they have allowed us to evaluate some of the software that is still under development. Mssrs. Will Druk and Mark Linnell from the Apple branch office in Salt Lake City have been very patient with our never-ending quest for information and they have spent valuable time and energy tracking down the answers for us. In our most recent conversations, we talked with them about the future, and most specifically about future of software development. Based on those conversations, we have put together a long list of software currently in the development process. We were also given some software to try out and analyze. By the time you read this, we expect all of these programs to be on the market. The actual programs that end up being sold may vary somewhat from our descriptions, as these programs were not in their final form when we got our hands on them. Also, there was no documentation available as yet so our descriptions of the program's functions are entirely the result of trial and error explorations. Since both of the authors of this book are frequently involved in the writing of documentation for new computer programs, we are well aware of last-minute changes or new directions in a developing application program. We therefore apologize, in advance, if our descriptions fail to do some programs full justice.

NEW SOFTWARE FOR THE MAC

Below are brief descriptions of software under development for use on the Apple Macintosh.

*MacEdge (By Think Educational Software)*

MacEdge is a group of educational programs for kids. The programs are divided into two folders, the Reading Folder and the Math folder. In each are a series of learning problems in game form that range from very easy to very complicated. The word games require matching and searching for word types. Some of the games are quite innovative and all take advantage of Macintosh’s mouse and menuing functions. Few keystrokes are required in any of the programs, which will probably seem like an advantage to the kids playing the games, but a disadvantage to parents and teachers who would like to teach some typing skills at the same time the kids are trying to figure out the games.

For example, the *Memory Match* game is a game that requires one to four players to reveal the words that hide in several boxes and then remember where the original word was hidden when a match is found. However, the player never has to actually type in any of the words, they just have to find them with the mouse pointer and click. There used to be a television matching game just like this game. Neither the computer game nor the television show is likely to teach anybody very much, but the program does go beyond the usual limitations of this game by adding new twists such as matching up words with their
contractions, matching antonyms, synonyms, homophones, and compound words. Successes are rewarded with random phrases such as "Great," "Where have you been hiding all this talent?" or "This is quite an accomplishment."

*Word Wonder* is another learning game that requires the players to find a word hidden in a matrix of letters. The word has to fit into a sentence. The word choices start out easy and don't ever get that much more difficult, so this is a game aimed at younger audiences.

In the Math Folder are six math-based learning games that range from ordinary to pretty good. The programs present a variety of math problems, and, unlike the word programs, let the user enter the numbers from the keyboard. Therefore, the programs could also be used as an easy and relatively nonthreatening introduction to the computer keyboard.

The first game, *Count On Mac* gives you two chances to guess the right number. This is a simple counting game for young kids that could also be used as an easy introduction to the Macintosh and the mouse pointer. All of the math programs include a pull-down help screen that explains in detail how to play the game.

*Additional Subtraction* expands on the counting program to give kids a chance to try simple addition and subtraction problems. Again, the program gives you two chances and then presents the correct answer.

*A L'il Give 'n Take* is another add and subtract program that uses computer games sound effects to reward success; stoney silence greets two incorrect answers.

*Give and Take* offers more addition and subtraction types of problems, while *Good Times* and *Dividing Line* expand the principle to multiplication and division problems. The latter two programs offer much more challenging games. For example, the player can specify tough division problems with variable dividend and divisor sizes or two numbers of varying sizes can be multiplied with or without carrying.

*Dollars and Sense*
(by Monogram, a division of Tronix Publishing)

We were given a "limited addition" version of *Dollars and Sense*, a program to keep track of accounts and budgets. In this case, limited edition doesn't mean that the program was only produced for a few select people, like a classic car; it really refers to the fact that not all the functionality of the program is included on the sample diskette. There is even a message in the program that invites you to make copies of the program and give them to your friends: an interesting marketing strategy.

Selecting NEW from the file menu presents the user with an "account card." This card makes it very clear how to enter an account: All you do is move the mouse pointer to the field where you want to enter data and key it in. Then the data is treated as a standard data base and the program provides a variety of ways to analyse the entries.

The best thing about the program is that it takes what can be a very confusing and difficult to learn program—the data base management program—and makes it a logical, visually-oriented program. The idea of ledger cards filed away somewhere makes sense to
all of us. The program simplifies and speeds up this time-honored method by giving the user a variety of new ways to analyze the data once the electronic cards are stored.

**MacProject**

*MacProject* is one of Apple’s most innovative and complex new programs. It is used to map the planning route and development milestones of an ongoing project. It allows you to easily draw and label task boxes and then to connect them to key dependent events that must occur in concert with these tasks. It includes a calendar that can be used to analyze problems with projected milestone dates and schedules. The program makes it easy to track dependent costs, schedules, and key dates in a sequence of project events. As an example, the program also includes a sample project, one in which a new site expansion plan is mapped out. Key milestones such as contacts with the realtor, visits to the property, investigation of zoning laws, and analysis of the area are all mapped and related to dates and other required tasks. The program seems like it could be useful not only as a planning tool but as a clear, graphic history of a project's development. For a project planner of any type, MacProject will allow the user to create a clear map of any project's history and then provide a variety of printouts that graphically show the project structure. Combined with the Macintosh’s graphics printer, the program makes creating charts for overhead projection a snap.

But the real value of the program is the many subprograms that provide several ways to analyze the data once it is entered on the charts. All of the project’s milestones are treated as entries in a data base and can be retrieved in a number of ways, each lending a different sort of analysis.

**Sargon III (by Hayden Software)**

This is an adaption of the popular “play against the computer” chess game that is also available for a variety of other brands of computers. It loads like any other finder-based program, but then immediately puts you into a chess game against the computer. You play the white pieces and the computer plays the black pieces. On the familiar pull-down menus are a number of options. You can choose the level of difficulty of play, which is expressed as the number of seconds you give the computer to “think” about each move. You, as the player, can take as much time as you want. What makes it more difficult, is that the longer the time limit the computer is allowed before it has to move, the more options it can look over before each move. Give the computer enough time to evaluate the situation and it becomes almost unbeatable.

Other options allow you to change the shades of the chessboard squares, to verify your moves, to switch colors, and so on. You can also ask the computer to show you some great games, or just to show you some opening or closing strategy. You can also save your partially played games on disk or print them out on a printer.

All in all, we found the Sargon chess program to be a pretty good chess player—if somewhat aggressive—and a good teacher of the game. There were several functions we never could get to work, probably because we didn’t have the documentation to explain it
to us. One that we would have liked very much was the *undo* function, which we assume meant that we could change our last move. That is, you can take back that last devastatingly clever move that results in the computer taking the bishop that you forgot about protecting.

**Word (by Microsoft)**

Microsoft's *Word*, a new word processing program, is probably a better choice for people who need a full-capability word processor than Apple’s own MacWrite program. Offering a large number of standard word processing functions, the program was under development even before the Macintosh was announced. In addition to the Mac's usual features such as the clipboard and the desktop tools, Word includes so many new options as to make it impossible to list them here. Suffice it to say it is a full-function word processor with all the usual formatting and printing, block handling, and on-screen text manipulation routines.

The copy of Word we had showed a discomfiting tendency to run into Bomb errors (serious system errors that result in a picture of a bomb appearing on the screen), but we had an early development copy of the program. Having used many Microsoft programs in the past, we know they are fully tested and bug free by the time they appear in the stores.

**Multiplan and Chart (by Microsoft)**

*Multiplan* and *Chart* are two different programs, but they work hand in hand. Microsoft’s Multiplan program has been around for a couple of years and it is one of the most popular of all the spreadsheet type programs. There is a version of Multiplan available for most of the business computers on the market. But never has there been a better implementation of this program than on the Macintosh. If you haven’t worked with spreadsheet-type programs you have missed one of the most valuable software applications programs for microcomputers. They can be used to prepare tables and charts, but they function best as a what if planning tool, especially if the planning involves numbers and requires testing more than one alternative. With a spreadsheet program like Multiplan, you can map out the general structure of the plan by entering into the spreadsheet cells the factors upon which a decision must be made. Then you can change the numbers in each of the cells to reflect what might (or might not) happen.

The key to the program’s usability is that every time you enter a new number in one cell, all the other cells that are dependent upon that one will also change. If you are calculating income versus expenses to discover profits, increasing costs will clearly show decreasing profits; increasing income will quickly be seen to offset costs and the cell in the spreadsheet that shows profits will almost instantly increase also. When the sources of income and expenses in such a project become many and varied, this kind of spreadsheet evaluation is about the only way to keep track of such a large number of ever-changing variables. A school course on financial planning that fails to look into electronic spreadsheets will certainly be missing the wave of the future.
This spreadsheet program for the Macintosh is about the best we have seen. To begin with, the general spreadsheet concept—an array of cells that can be used to hold labels, numbers, or formulas—seems to be particularly well suited to the use of the mouse. By just sliding the mouse across the tabletop, you can move from cell to cell much faster than you could with the usual cursor keys. The mouse also lets you select a group of cells very quickly just by holding down the mouse button and sliding the mouse across the cells you want to work with.

Microsoft's Chart is a logical companion to Multiplan because it gives the user a way to display graphically the results of Multiplan's calculations. It is a combination graphics drawing program and a way to organize and make meaningful sense of the extended calculations that can be developed using Multiplan. Most valuable is its capability to take data from existing Multiplan spreadsheets and chart them.

**MacTerm**

*MacTerm* is Apple's new terminal emulation and modem program. It includes all the usual features such as the desktop accessories, pull-down menus, and click boxes. But it is more than a computer applications program; it allows the Macintosh to take on the characteristics of other types of terminals. The pull-down menus offer choices such as UT100-type terminal emulation or TTY emulation. You can set the operating mode, and specify such things as line width, nonstandard character sets, a variety of on-line choices, and even the size and shape of the cursor. The pull-down menus allow you to set the terminal type and options, the Compatibility, File Transfer, and Answerback options and a number of other features that left us wishing that the documentation for this program was ready. There were a number of features that had us guessing as to their purpose. And the MacTerm disk we were given came with a group of programs on it that seemed to be related to other functions besides terminal emulation. Such titles as Apple Cluster Controller, Hard Disk Commercial Services, AppleLine Supervisor, and Mac to Mac left us guessing. We suppose we will have to wait along with everybody else to learn what all those things do.

**Think Tank (From Living Videotext, Inc.)**

*Think Tank* is not a complicated program, but that's the beauty of it. It's the kind of program you might not expect to have much use for, and then end up using it a lot. When we visited the independent software coordination group at Apple headquarters in Cupertino, California, we heard about Think Tank. Guy Kawasaki, at Apple, spoke so highly of it we decided to send away for our own copy. The people at Living Videotext were nice enough to send us a free copy.

When we first loaded the Think Tank program, we were not especially impressed. Compared to the large number of options available on some of the other programs we were testing, the top of the screen—where the pull-down menus with all the program's options are hidden—was relatively sparse. In addition to the usual Accessories, File, and
Edit menus, there was only one menu called Extra, and one for Fontsize. Pull down the Fontsize menu and you get your choice of 9 point or 12 point type. Clearly, this program is not out to dazzle you with fancy printing on the screen. Pull down the Extra menu and you get three choices: Search, Sort, and Status. It doesn’t take long to discover that this is a working program, not a bells and whistles show off. The fact is, Think Tank doesn’t do all that much, but what it does it does well. Think Tank is for all those people who like to organize their thoughts on paper, or, in this case, on the screen. It is set up as an outlining program, but that is only the organizing principle; it’s really a program that allows a way to get your thoughts down in black and white. That way they can be viewed, organized, and easily modified.

The basic principle is that you can organize your ideas into headlines and subheads. It’s not a new idea; outlining as a way to organize thoughts has been suggested by many thinkers. But this is the first time a program has been written that allows us to use a computer in the process. The program not only guides and speeds the creation of the outline, but it has a way to hide the various levels of subheadings so you only have to ponder the main topics. The subtopics can be recalled when they are relevant. All you have to do is move the mouse pointer to a main heading and click; the subtopics under that heading instantly appear. Click again and the subheadings are hidden. The choices in the Extra menu allow you to quickly resort your subheadings or to search for a particular word or phrase. The Status option reports the name of the file you are working on and tells you how much memory you have used up. If your outline grows beyond the size of the screen, you can scroll the text in any direction just by moving the mouse pointer to the edge of the box and pressing; you don’t even have to look for scroll bars on this one. It’s just one example of the slick implementation of the program. It fits the Mac environment well.

That’s really about it. There are some other functions, but the outlining approach gives the main value of the program. Of course, you can save your outlines on diskette and/or print them out on an attached printer. If you are entering text and don’t want to take your hands off the keyboard to manipulate the mouse, there is a way to do most of the program’s functions by pressing combinations of keys. The best thing about this program is that it is easy to use and it does what it is supposed to do.

MORE FUTURE SOFTWARE

We wish we could wait just a bit longer before going to press with this book. So much new software is appearing every day it would be interesting (and educational) to keep trying out all those new programs. But we always feel that way when it is time to put a book to bed; there are always a dozen things that we could have done if only we had had a few weeks more. But this time it is especially difficult. It is not only that the Macintosh is a brand new computer and so new software is appearing every day. It’s also tied up with the nature of this new machine. It’s more than just a new computer, it’s a new way to use computing power. We are fascinated by the interesting new types of programs that are being stimulated by this new generation computer. But, we will just have to be patient, along with everybody else. There will be interesting adaptations of existing software forth-
Macintosh Software

coming too. Many of them will find innovative ways to work hand in hand with the Mac­
intosh’s new approach to user interaction. There will also be new types of programs that,
despite the best predictions of the experts, we can’t even imagine now.

Apple has shown the way with their innovative programs like MacPaint and
MacDraw, with MacProject, and with the desk accessory programs. Now the task has
been turned over to the independent software producers. We are eager to see what they
will come up with next.

SOFTWARE UNDER DEVELOPMENT

When we visited Macintosh headquarters in Cupertino, California, we were given an in­
side glimpse of the support that new software developers are being given by the Apple
crew. We were also handed a list of software that is currently being created for the Mac.
That list is reproduced below.

Aardvark/McGraw-Hill
738 N. Water St.
Milwaukee, WI 53202
(414)289-9988

Estate Tax Planner Taxes
Personal Tax Planner Taxes
Professional Tax Planner Taxes

Accountants Microsystems
1404 140th Place NE
Bellevue, WA 98007
(206)643-2050

Personal Tax Machine Taxes
Tax Machine Taxes

Advanced Data Institute
1215 Howe Avenue
Sacramento, CA 95825
(916)925-2229

Aladin Database

AgDisk/Harris Technical Systems
624 Peach St.
PO Box 80837
Lincoln, NB 68501
(800)228-4091
(402)478-2811

Banking Products Agricultural
Budgeting Agricultural
Business Management Agricultural
Cash Flow Agricultural
<table>
<thead>
<tr>
<th>Category</th>
<th>Program Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corn/Soybean Management</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Cow—Calf Management</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Crop Management</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Crop Recordkeeping</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Enterprise Analysis</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Farm Accounting</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Feedlot Cattle Management</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Financial Management</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Inventory and Position</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Machinery Management</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Market Charting</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Profit Projector</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Swine Far. Management</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Swine Finishing Management</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Swine Management</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>Swine Recordkeeping</td>
<td></td>
<td>Agricultural</td>
</tr>
<tr>
<td>American Intellware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>330 Washington St. 4th Floor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marina Del Ray, CA 90292</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(213)827-0803</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacFacts</td>
<td></td>
<td>General</td>
</tr>
<tr>
<td>Ann Arbor Softwords</td>
<td></td>
<td></td>
</tr>
<tr>
<td>308½ South State Suite 30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ann Arbor, Michigan 48104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(313)996-3838</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-D Graphics</td>
<td></td>
<td>Graphics</td>
</tr>
<tr>
<td>Animation Toolkit 1</td>
<td></td>
<td>Graphics</td>
</tr>
<tr>
<td>Project/Time Management</td>
<td></td>
<td>Project Management</td>
</tr>
<tr>
<td>Apple Computer, Inc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20525 Mariani Avenue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cupertino, CA 95014</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(408)996-1010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td></td>
<td>Game</td>
</tr>
<tr>
<td>MacBasic</td>
<td></td>
<td>Language</td>
</tr>
<tr>
<td>MacDraw</td>
<td></td>
<td>Graphics</td>
</tr>
<tr>
<td>Macintosh Assembler/Debugger</td>
<td></td>
<td>Language</td>
</tr>
<tr>
<td>Macintosh Logo</td>
<td></td>
<td>Language</td>
</tr>
<tr>
<td>Macintosh Pascal</td>
<td></td>
<td>Graphics</td>
</tr>
<tr>
<td>MacPaint</td>
<td></td>
<td>Project Management</td>
</tr>
<tr>
<td>MacProject</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applied Software Technology</td>
<td></td>
<td></td>
</tr>
<tr>
<td>170 Knowles Drive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Los Gatos, CA 95030</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(408)370-2662</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VersaForm</td>
<td></td>
<td>Database</td>
</tr>
</tbody>
</table>
Apropos Software
64 Hillview Drive
Los Altos, CA 94022
(415)948-7227
Auto Purchase/Lease Spreadsheet
Education Fund Planner Spreadsheet
Home Budget Spreadsheet
Life Insurance Planner Spreadsheet
Loan Planner Spreadsheet
Personal Financial Statement Spreadsheet
Real Estate Property Planner Spreadsheet
Stock Portfolio Planner Spreadsheet

Ashton-Tate
101150 W. Jefferson Blvd.
Culver City, CA 90230
(213)204-5570
dBase II Database
Friday! Database

Axion
1287 Lawrence Station Rd.
Sunnyvale, CA 94089
(408)747-1900
Art Portfolio Graphics

Banbury Books, Inc.
353 W. Lancaster Avenue
Wayne, PA 19087
(215)964-9103
Typing Tutor Education

Blue Chip Software
6744 Eton
Canoga Park, CA 91303
(818)346-0730
Baron Game
Millionaire Game
Squire Game
Tycoon Game

Borland International
4113 Scotts Valley Drive
Scotts Valley, CA 95066
(408)438-8400
Turbo C Language
Turbo Modula Language
Turbo Pascal Language
BPI Systems, Inc.
3423 Guadalupe
Austin, TX 78705
(512)454-7191

Accounts Payable    Accounting
Accounts Receivable Accounting
General Accounting  Accounting
Payroll             Accounting

Bristlecone Systems Development
742 West 200 South
Cedar City, UT 84720
(801)586-5050

QuickFile          Database
QuickCom           Communication

Brock Software Products, Inc.
PO Box 799
Crystal Lake, IL 60013
(815)459-4210

Keystroke Data Base Database
Keystroke Report Base Database

Broderbund
17 Paul Drive
San Rafael, CA 94930
(415)479-1170

Load Runner        Game

Business & Professional Software
143 Binney St.
Cambridge, MA 02142
(617)491-3377

Art Department     Graphics

CBS Software
One Fawcett Place
Greenwich, CT 06836
(203)622-2615

Murder by the Dozen Game

CESI
8600 NW South River St. 212
Miami, FL 33166
(305)887-8217

Electronic Rolodex Database
Challenger Software  
2927 Balmoral  
Flossmoor, IL 60422  
(312)957-3475

  Legacy  
  Music Composition  
  Printing Cost Estimator  
  Game  
  Music  
  Accounting

Chang Laboratories, Inc.  
5300 Stevens Creek Blvd.  
Suite 200  
San Jose, CA 94901  
(408)246-8020

  Consolidation Module  
  DocuPlan  
  FilePlan  
  GraphPlan  
  Link Module  
  MemoPlan  
  MicroPlan  
  Spreadsheet  
  Spreadsheet  
  Database  
  Graphics  
  Spreadsheet  
  Database  
  Spreadsheet

Computer Poet, Inc.  
999 Tahoe Blvd.  
P.O. Box 7707  
Incline Village, NV 89450  
(702)831-7440

  Computer Poet  
  Education

Continental Software  
11223 S. Hindry  
Los Angeles, CA 90045  
(213)410-3977

  Home Accountant  
  Small Business Accounting  
  Accounting  
  Accounting

Creative Solutions, Inc.  
4701 Randolph Road  
Rockville, MD 20852  
(301)984-0262

  Graphic Terminal Emulator  
  MacForth I  
  Communications

DataPak Software, Inc.  
14755 Ventura Blvd.  
Sherman Oaks, CA 91403  
(818)905-6419

  MacJack  
  Game
DeskTop Software Corp.
228 Akeabder St.
Princeton, NJ 08540
(609)924-7111

1st Base
Graph-N-Calc

Digital Research Inc.
160 Central Avenue.
Pacific Grove, CA 93950
(800)547-1842

Dr. Logo

Dilithium Software
8285 Nimbus, Suite 151
Beaverton, OR 97005
(800)547-1842

Brain Games
Comquest
Mac Tutor
PC to Mac and Back
Telofacts I
Telofacts II

Dow Jones & Co., Inc.
US Route 1 and Ridge Road
South Brunswick, NJ 08852
(609)452-2000

Dow Jones Market Manager
Dow Jones Spreadsheet Link
Dow Jones StraightTalk

Execucom
PO Box 9758
Austin, Texas 79766
(512)346-4980

Mac IFPS/Personal

Execuware
7415 Pineville-Mathews Road
Charlotte, NC 28226
(800)438-3636

Executive Productivity Tools

Fingertip Software
3327 E. 14th Street
Long Beach, CA 90804
(213)498-8064

500 Menu Patterns for Macpaint
Future Novelties, Inc.
P.O. Box 1412
214 Arlington St.
Rocky Mount, N.C. 27801
(919)442-9510

Aviation Education
Money Market Game
States and Capitals Education

Great Plains Software
1701 38th SW
Fargo, ND 58103
(701)281-0550

Accounting Series Accounting

Haba Systems
15154 Beacon Street
Van Nuys, CA 91405
(818)901-8828

Habadex for Mac Calendar
Habawindow Calendar Calendar

Harvard Associates, Inc.
260 Beacon St.
Somerville, MA 02143
(617)492-0660

Educational Simulation Game Education
MacManager Game

Hayden Software Co.
600 Suffolk St.
Lowell, MA 01853
(800)343-1218

Butter Spreadsheet
DeVinci Game
Sargon III Game
The Calendar Calendar
The List Manager Database
The Planner General

Hayes Microcomputer Products
5923 Peachtree Ind. Blvd.
Norcross, Georgia 30092
(404)449-8791

Communications System Communication
Health Enhancement & Promotion  
P.O. Box 546  
Ames, Iowa 50010  
(515)233-3552

   MacHealth I      General
   MacHealth II     General

Horizon Software Systems  
185 Berry St. 4821  
San Francisco, CA 94107  
(415)543-1199

   Horizon Spreadsheet  Spreadsheet
   Horizon Word Processing  Word Processing

Human Edge Software Corp.  
2445 Faber Place  
Palo Alto, CA 94303  
(415)493-1593

   The Communications Edge  General
   The Leadership Edge       General
   The Management Edge       General
   The Negotiation Edge       General
   The Sales Edge             General

Infocom  
55 Wheeler St.  
Cambridge, MA 02138  
(617)492-1031

   Deadline                  Game
   Enchanter                  Game
   Infidel                    Game
   Planet Fall                Game
   Starcross                  Game
   Witness                    Game
   Zork I, II, and III        Game

Kriya Systems, Inc.  
505 North Lakeshore Drive  
#5510  
Chicago, IL 60611  
(312)822-0624

   Business Simulation       Education
   Speed Reading              Education

Learningways, Inc.  
98 Raymond St.  
Cambridge, MA 02140  
(617)576-3007

   Idea Processor            Education
Living Videotext Inc.
2432 Charleston Rd.
Mountain View, CA 94043
(415)964-6300

Think Tank

Logic Systems
1009 So. 5th St.
St. Charles, IL 60174
(312)584-4548

Label-It
Picture-Disk

Logo Computer Systems, Inc.
9960 Cote de Liesse
Lachine, Quebec HBT 1A1
(514)631-7081

Macintosh LOGO

LOKI Engineering Inc.
55 Wheeler St.
Cambridge, MA 02138
(617)576-0666

Hearts
Mac MAGICAL Language
Webster’s Revenge

Lotus Development Corporation
161 First St.
Cambridge, MA 02142
(617)494-1270

Lotus

Macinsoft
PO Box 27583
San Diego, CA 92127
(619)745-6084

PaintMover

Magnum Software
21115 Devonshire St.
Chatsworth, CA 91311
(818)700-0510

McPic
Main Street Software
1 Harbor Dr.
Sausalito, CA 94956
(415)332-1274
Mainstreet Filer General
Mainstreet Writer Word Processing

Mark of the Unicorn
222 3rd St.
Cambridge, MA 02142
(617)864-2676
Mouse Stampede Game
Professional Composer Music

Megahaus Corp
5703 Oberlin Drive
San Diego, CA 92121
(800)451-1230
MegaCalendar Calendar
MegaDesk General
MegaFinder Database
MegaFile Database
MegaForm Database
MegaMerge General
MegaSpell General

Micro Focus, Inc.
2465 East Bayshore Road
Suite 400
Palo Alto, CA 92648

COBOL Language

Microfacts, Inc.
5401 W. Kennedy Blvd.
Suite 632
Tampa, FL 33609
(813)876-4287

Desk Manager Database
Calendar Calendar

Microrim, Inc.
1520 112th NE
Bellevue, WA 98004
(206)453-6017

Database Management System Database
Microsoft
10700 Northup Way
Bellevue, WA 98004
(206)828-8080

Microsoft BASIC       Language
Microsoft Chart       Graphics
Microsoft File        Database
Microsoft Word        Word Processing
MultiPlan             Spreadsheet

Miles Computing, Inc.
7136 Sixth Avenue
Van Nuys, CA 91406
(818)994-7901

MacAttack             Game
Mac the Knife         Graphics
Mac the Knife II      Graphics

MECC
Minnesota Ed. Computing Consortium
3490 Lexington Avenue North
St. Paul, Minnesota 55112
(612)481-3569

Three Dimensional Graphics        Graphics

Monogram
8295 S. La Cienaga Blvd
Inglewood, CA 90301
(213)215-0529

Dollars and Sense          General

Oasis Systems
7907 Ostrow
San Diego, CA 92111
(619)279-5711

Sundog                    Game

Open Systems
430 Oak Grove
Minneapolis, MN 55403
(612)870-3515

Accounts Payable          Accounting
Accounts Receivable        Accounting
General Ledger              Accounting
Job Cost                    Accounting
Inventory                  Database
Payroll                     Accounting
Purchase Order  General
Report Writer  General
Sales Order Entry  General
SMA Business Basic  General

Orion Software
38 Dew Drop Road
York, PA 17403
(717)757-7721

Checkwriter  General
The Master  General

Palantir Software
3400 Montrose Blvd.
Suite 718
Houston, TX 77006
(713)520-8221

Palantir Word Processing  Word Processing
MacType  Education

Penguin Software
830 4th Avenue
Geneva, IL 60134
(312)232-1984

Pensate  Game
The Coveted Mirror  Game
The Graphic Magician  Game
The Quest  Game
Transylvania  Game
Xyphus  Game

Sierra On-Line
Sierra On-Line Building
Coarsegold, CA 93614
(209)683-6858

Frogger  Game

Simon and Schuster
1230 Ave of Americas
NY, NY 10023

Typing Tutor  Education

Simple Software
220 Redwood Highway
Mill Valley, CA 94941
(415)381-2650

Checkbook  Accounting
Macintosh Software

Sir Tech
6 Main Street
Ogdensburg, NY 13669
(315)393-6633
Wizardry Game

Sirius Software, Inc
10364 Rockingham Dr.
Sacramento, CA 95827
Snake Byte
Type Attack Game
Education

Soft-Life Corporation
2950 Feliz
LA, CA 90039
(213)660-7940
MacSlots Game

Softech Microsystems
16885 W. Bernardo Dr
San Diego, CA 92127
(619)451-1230
Fortran 77 Dev Sys
UCSD Pascal Dev Sys Language
Language

Software Arts, Inc.
27 Mica Lane
Wellesley, MA 02181
(617)237-4000
TK! Solver General

Software Publishing Corp.
1901 Landings Dr.
Mountain View, CA 94043
(415)962-0191
PFS: FILE General
PFS: REPORT General

Softworks Limited
607 W. Wellington
Chicago, IL 60657
(312)975-4030
APL Language
Assembly Language
C Compiler Language
Fortran Language
Southwest Software
7743 Briarwood Dr
New Orleans, LA 70128
(504)246-8438

MacTransfer

State of the Art, Inc.
3183 Building A Airway Ave.
Costa Mesa, CA 92626
(714)850-0111

MAC General Accounting

Stoneware
50 Belvedere St.
San Rafale, CA 94901
(415)454-6500

DB Master

Superex Intl Marketing LTD
151 Ludlow St.
Yonkers, NY 10705
(800)862-8800

Home Executive

T/Maker Company
2115 Landings Dr.
Mountain View, CA 94043
(415)962-0195

ClickArt

Telos Software Products
3420 Ocean Park Blvd
Santa Monica, CA 90405
(213)450-2424

Filevision

The Original Computer Camp
559 San Ysidro
Santa Barbara, CA, 93108
(805)969-7871

LISP

LOGO
Think Educational Software
16 Market St.
Potsdam, NY 13676
(315)265-5636

MacEdge
Mind Over Mac

Education
Games

TOM Software
PO Box 66596
Seattle, WA 98166
(206)246-7022

Property Management
Public Accountant

Real Estate
Accounting

Unify Corporation
3333 Bowers Ave.
Santa Clara, CA 95051
(408)727-1188

Unify Relational DB

Data Base

UniPress Software
1164 Raritan Ave.
Highland Park, NJ 08904
(201)985-8000

ADA
C Development
COBOL
Lex Word Processor
SVS BASIC-Plus
SVS FORTRAN
SVS Pascal
Unicalc
UNIX V.5

Language
Language
Language
Word Processing
Language
Language
Language
Spreadsheet
Language

Vested Interest Programs
1356 Thunderbird Avenue
Sunnyvale, CA 94087

Stock Market Investing

Stock Market

Videx, Inc.
1105 Northeast Circle
Corvalis, OR 97330
(503)758-0521

Mac Desktop Calendar

Calendar

Scarborough Systems, Inc.
25 North Broadway
Tarrytown, NY 10591
(914)332-0445

Run for the Money

Game
Sentient Software, Inc.
1280 Ute Avenue
Aspen, CO 81611
(303)925-9293

CYBORG

Quadratron
15760 Ventura Blvd.
Suite 1032
Encino, CA 91436
(213)789-8588

Q-Calc Calculator General
Q-Call Phone Directory Communication
Q-Date Calendar Calendar
Q-Form Screen Builder General
Q-Mail Electronic Mail Communication
Q-Menu Utility Utility
Q-Note Card General
Q-Office General
Q-One Word Processor Word Processing

Real World Corporation
Water View Plaza Ste. 103
Parsippany, NJ 07054
(800)225-1115

Accounts Payable Accounting
Accounts Receivable Accounting
General Ledger Accounting
Order Entry, Inventory Accounting
Payroll Accounting
Sales Analysis Database

REMS Software
526 NW Second St.
Corvallis, OR 97330
(503)757-8887

Real Estate Appraiser 1004 Real Estate
Real Estate Investor I & II Real Estate
Real Estate Rent Vs Buy Real Estate
The Financial Package Real Estate

Roger Wagner Publishing
10761 Woodside Avenue
Suite E
Santee, CA 92071
(619)562-3222

Typing Training Education
Utilities Utility
A

EDUCATIONAL COMPUTER MAGAZINES AND JOURNALS

ACM SIGCUE Bulletin
Computer Uses in Education
Association for Computing Machinery
P.O. Box 12015
Church Street Station
New York, NY 10249

Articles, reviews, and information resources about computer-based instructional materials. Reports on research about the value of computer-aided instruction and offers interviews with leaders in educational computing.

Access: Microcomputers in Libraries
P.O. Box 764
Oakridge, OR 97463
(quarterly/$11)

Features articles on microcomputer applications for libraries. Includes software reviews.

AEDS Journal and AEDS Monitor
Association for Educational Data Systems
1201 16th Street NW
Washington, DC 20036
(quarterly)
The AEDS Journal provides reports on original research and theoretical articles related to educational computing, often focusing on problems in instructional design and administrative applications. The AEDS Monitor (bimonthly) includes research and reviews from other organizations such as ERIC and MECC.

Classroom Computer Learning
19 Davis Drive
Belmont, CA 94002
(nine per year)

For precollege educators. Features articles on using the computer in the classroom. Separate sections for elementary, middle, and high school. Includes software reviews and new product information.

Classroom Computer News
Intentional Educations
51 Spring St.
Watertown, MA 02171
(bimonthly/$12)

Features aspects of classroom computers and educational technology in easy-to-understand language. Includes teacher-developed classroom applications and original programs.

Collegiate Microcomputer
2706 Wilson Drive
Terre Haute, IN 47803
(812)877-5511
(quarterly/$28)

Articles provide information on applications for teaching and research with microcomputers in the classroom, laboratory, and library.

Computers, Reading and Language Arts
Berkeley Enterprises, Inc.
Box 13247
Oakland, CA 94661-0247
(415)339-1106
(quarterly/$14)

Features articles on computer-aided instruction for precollege educators of reading and language arts. Reviews of books and software.

Computer-Using Educators Newsletter
Independence High School
1776 Education Park Dr.
San Jose, CA 95133
(bimonthly)
Computing Teacher
International Council for Computers in Education
University of Oregon
Eugene, OR 97403
(503)686-4429 or 686-4414
(Monthly/$16.50)

A magazine for precollege educators. Features articles on software and programming for educational applications. Regular columns provide information on computers for teaching English and science. Book and software reviews included.

Courseware Magazine
4919 N. Millbrook, #222
Fresno, CA 93726
(five per year)

Educational Computer Magazine
P.O. Box 535
Cupertino, CA 95015
(408)252-3224
(ten per year/$25)

This magazine is for teachers, administrators, and media specialists who are interested in using computers in education. Hardware and software reviews are featured.

Educational Computing
MAGSUB (Subs. Services) Ltd.
Oakfield House
Perrymount Road
Haywards Heath, Sussex RH16 3DH
England

A British magazine that reports on instructional applications in schools, colleges, and universities. Provides a good perspective of British views on educational computing.

Educational Technology
Educational News Service
140 Sylvan Avenue
Englewood Cliffs, NJ 07632
(bimonthly/$12)

Electronic Education
Electronic Communications, Inc.
Suite 220
1311 Executive Center Dr.
Tallahassee, FL 32301
Tel: (904)878-4178
(ten per year/$15)

Provides nontechnical information about the educational use of computers. Includes reviews of computer systems and software.
Electronic Learning
902 Sylvan Avenue
Englewood, NJ 07632
(212)505-3000
(Six per year/$19)

Designed for educators and educational administrators. Features articles on the applications and advances of technology in the classroom. Includes software reviews and new product information.

Instructional Innovator
Association for Educational Communications & Technology
1126 16th St. NW
Washington, D.C. 20036
(202)466-4780
(eight per year/$24)

Will keep you up to date on media technology for educational purposes.

Instructor Computer Director for Schools
757 Third Avenue
New York, NY 20036
(212)888-3400
(annual/$19.95)

For precollege educators. This annual publication provides a guide to microcomputers, including categories of hardware, software, publications, and companies.

Interface: The Computer Educational Quarterly
Stephen Mitchell, Publisher
915 River Street
Santa Cruz, CA 95060
(408)425-3851
(quarterly/$11)

A journal for computer science and data professionals that includes articles on instructional uses of computers.

Journal of Computer-Based Instruction
ADCIS
409 Miller Hall
Western Washington University
Bellingham, WA 98225
(quarterly)

Features theoretical articles, lectures, and reports on research on computer-based instruction in elementary and secondary schools, colleges, business, the military, and government agencies.
Educational Computer Magazines and Journals

Journal of Computers in Mathematics and Science Teaching
P.O. Box 4455
Austin, TX 78765
(512)258-8083
(quarterly/$15)

Provides a forum for teachers of mathematics and science to exchange ideas regarding the use of microcomputers in the curriculum. Features reviews of math and science software and has a calendar of conferences and events of interest.

Journal of Computers, Reading and Language Arts
(CRLA)
NAVA (National Audio Visual Association)
3150 Spring St.
Fairfax, VA 22031

Journal of Educational Technology Systems
Baywood Publishing Company, Inc.
120 Marine Street, Box D
Farmingdale, NY 11735
(516)240-7130
(quarterly)

A technical journal primarily concerned with curriculum and program development. It is designed for those who are developing curriculum projects for education.

Logo and Educational Computing Journal
Suite 219
1320 Stony Brook Road
Stony Brook, NY 11790

MACWORLD
PC World Communications, Inc.
555 DeHaro St.
San Francisco, CA 94107

Features articles, reviews, and special information for Mac users. Includes special section on uses of the Mac with the Mac Paint program.

Media and Methods
Society of Educators
1511 Walnut St.
Philadelphia, PA 19102
(215)563-3501
(nine per year/$24)

A magazine to keep teachers up-to-date on the latest media technology for the classroom. Features articles and special columns on microcomputers, software, and applications for teaching.
Microcomputers in Education
5 Chapel Hill Dr.
Fairfield, CT 06432
(203)335-0908
(monthly/$38)

*This newsletter is designed for educators at all levels. It offers the educator software and hardware reviews, book reviews, industry developments, and a calendar of events.*

National LOGO Exchange
Box 5341
Charlottesville, VA 22905
(nine per year/$25)

*This publication is for teachers using LOGO. Find out the latest methods and applications being used with this programming language. Includes programming suggestions and new product announcements.*

Pipeline
PO Box 388
Iowa City, IA 52244
(319)353-5789
(twice a year)

*Published by Conduit, Pipeline features articles on educational technology, pedagogy, and curriculum content.*

PLATO
Control Data HQA
8100 24th Avenue S.
Box O
Minneapolis, MN 55440
(quarterly)

*Valuable for those who are using PLATO and others interested in computer-based instruction.*

School Microcomputer Bulletin
Learning Publications, Inc.
Box 1326
Holmes Beach, FL 33509
(twice a month/$48)

*A new publication which will deal with trends in educational-computing concepts and examine commercial packages.*

St. Mac
Softalk Publishing
P.O. Box 7041
North Hollywood, CA 91605

*Articles, reviews, and special information for Mac users.*
Teaching and Computers
Scholastic Inc.
902 Sylvan Avenue
Box 2001
Englewood Cliffs, NJ 07632
(212)505-3000
(eight per year/$19)

Specifically designed for the elementary school teacher. Offers articles on methods of CAI and how to integrate the computer into the classroom. Includes a calendar of events, software reviews.

T.H.E. Journal
Information Synergy, Inc.
P.O. Box 17239
Irvine, CA 92713
(eight per year/$15)

For all educators. Each month articles cover a specific topic, such as technological innovations, as they pertain to education. Includes a calendar of events, software and hardware reviews, and new product announcements.
Basic for Beginners
Gary G. Bitter
1221 Avenue of the Americas
New York, NY 10020

The BASIC Handbook: An Encyclopedia for the BASIC Computer Language
David A. Lien
Compusoft Publishing, 1978
1050 Pioneer Way, Suite E.
El Cajon, CA 92020

BASICally Speaking: A Young Person's Guide to Computing
Frances Lieberman Cohen
11480 Sunset Hills Rd.
Reston, VA 22090

Before You Buy a Computer
Dona Z. Meilach
1 Park Avenue
New York, NY 10016
Best of Creative Computing
Vols. I-III
David Ahl and Burchenal Green, eds.
Creative Computing Press
39 E. Hanover Ave.
Morris Plains, NJ 07950

Best of Micro, Vols. II and III
Micro Link, Inc.
34 Chelmsford Street
PO Box 6502
Chelmsford, MA 01824

Com-Lit: Computer Literacy for Kids
Carin E. Horn and Carroll L. Collins
Sterling Swift Publishing Co., Inc., 1983
7901 South IH-35
Austin, TX 78744

Computer Camp Book
Laura E. Littel and Frederick D. Hickler, eds.
Yellow Springs Computer Camp, 1983
P.O. Box 292
Yellow Springs, OH 45387

Computer Consciousness: Surviving the Automated Eighties
H. Dominic Covvey and Neil H. McAlister, eds.
Addison-Wesley Publishing Co., Inc., 1980
One Jacob Way
Reading, MA 01867

Computer Literacy for School Administrators and Supervisors
Stephen Radin and Harold M. Breenberg
D. C. Heath & Company, 1983
Lexington, MA

Computer Literacy: Problem-Solving with Computers
Carin E. Horn and James L. Poirot
7901 South IH-35
Austin, TX 78744

Computer Programming for Young Children: A Step-by-Step Guide for Teachers & Parents
Cheryl Weinstein and Carol Harris
Creative Learning Press, Inc., 1983
P.O. Box 320
Mansfield Center, CT 06250
Computer Town: Bringing Computer Literacy to Your Community
Liza Loop, Julie Anton and Ramon Zamora
11480 Sunset Hills Road
Reston, VA 22090

Computer Tutor: An Introduction to Computers
Sandra Markle
Learning Works, Inc., 1981
P.O. Box 6187
Santa Barbara, CA 93111

Computers and Your Child
Ray Hammond
Tickner and Fields, 1984
52 Vanderbilt Avenue
New York, NY 10017

Computers for Everybody
Jerry Willis and Merl Miller
Dilithium Press, 1981
8285 S.W. Nimbus Street
Beaverton, OR 97005

Computers in Mathematics: A Sourcebook of Ideas
Creative Computing Press
Dept. C028
39 E. Hanover Ave.
Morris Plains, NJ 07950

Computers in Today's World
Gary G. Bitters
John Wiley and Sons, Inc., 1984
605 Third Avenue
New York, NY 10158

Computers, Teaching & Learning
Jerry Willis and D. Lamont Johnson
Dilithium Press, 1983
8285 S.W. Nimbus St.
Beaverton, OR 97005

Courseware in the Classroom: Selecting, Organizing, and Using Educational Software
Ann Lathrop and Bobby Goodson
Addison-Wesley Publishing Co., Inc., 1983
One Jacob Way
Reading, MA 01867
Educational Technology: Readings in Programmed Instruction
John P. DeCecco, ed.
Holt, Rinehart & Winston, Inc., 1964
383 Madison Avenue
New York, NY 10017

Every Kid's First Book of Robots and Computers
David Thornburg
Compute! Publications, Inc., 1982
P.O. Box 5406
Greensboro, NC 27403

Everything You Always Wanted to Know about Personal Computers but Didn't Know How to Ask
Bruce Brown
The Devin-Adair Co., Inc., 1983
143 Sound Beach Ave.
Old Greenwich, CT 06870

Exploring with Computers
Gary G. Bitter
Julian Messner, 1981
New York, NY 10020

Games, Graphics & Sound
Ray Curnow and Susan Curran
Simon & Schuster, Inc., 1984
1230 Ave. of the Americas
New York, NY 20020

Growing Up with Computers: A Parents' Survival Guide
Frederick and Victoria Williams
William Morrow & Co., Inc., 1983
105 Madison Avenue
New York, NY 10016

Intelligent Schoolhouse: Readings on Computers & Learning
Dale Peterson, ed.
11480 Sunset Hills Rd.
Reston, VA 22090

Introduction to Computer Animation
Nat Wadsworth
Hayden Book Co., Inc., 1983
Fifty Essex St.
Rochelle Park, NJ 07662
Learning with Computers
Alfred Bork
Digital Press, 1981
12 Crosby Drive, E/44
Bedford, MA 01862

Learning with Your Home Computer
Ray Curnow and Susan Curran
Simon & Schuster, Inc., 1984
1230 Ave. of the Americas
New York, NY 10020

Microcomputers & The Three R’s
Carol Doerr
Hayden Book Co., Inc., 1979
Fifty Essex St.
Rochelle Park, NJ 07662

Nancy A. Watson
College of Education, Payne B-47
Arizona State University
Tempe, AZ 85287

Microcomputers in K-12 Education, First Annual Conference Proceedings
Pierre Barrette, ed.
Computer Science Press, 1982
Rockville, MD

Microcomputers in the Schools
James L. Thomas, ed.
The Oryx Press, 1981
2214 North Central Ave.
Phoenix, AZ 85004

Mindstorms: Children, Computers and Powerful Ideas
Seymour Papert
10 E. 53rd St.
New York, NY 10022

The Mind Tool: Computers and Their Impact on Society
Neill Graham
West Publishing Co., 1980
P.O. Box 3526
St. Paul, MN 55165
Musical Applications of Microcomputers
Hal Chamberlin
Hayden Book Co., Inc., 1983
Fifty Essex Street
Rochelle Park, NJ 07662

My Students Use Computers: Computer Literacy in the K-8 Curriculum
Beverly Hunter
11480 Sunset Hills Rd.
Reston, VA 22090

Organizing a Computer Club for Elementary School Children
John T. Riley and Judie L. Hurtz
Computer Directions for Schools, 1983
P.O. Box 1136
Livermore, CA 94550

Organizing Your Computer Program: Lab vs Classroom Usage
John T. Riley and Judie Hurtz
Computer Directions for Schools, 1983
P.O. Box 1136
Livermore, CA 94550

Parent's Guide to Personal Computers & Software
Consumer Guide Editors
Simon & Schuster, Inc., 1983
1230 Ave. of the Americas
New York, NY 10020

Picture This! An Introduction to Computer Graphics for Kids of All Ages
D. Thornburg
Addison-Wesley Publishing Co., Inc., 1982
One Jacob Way
Reading, MA 01867

Practical Guide to Computers in Education
Peter Kelman, ed.
Addison-Wesley Publishing Co., Inc., 1982
One Jacob Way
Reading, MA 01867

Questions & Answers on Word Processing
Peter Lafferty
Focal Press, 1982
10 Tower Office Pk.
Woburn, MA 01801
Run: Computers in Education
Dennis D. Harper and James H. Stewart
Wadsworth Publishing Co., Inc., 1983
10 Davis Drive
Belmont, CA 94002

Student Involvement—Implementing a Computer Tutor Program
John T. Riley and Judie L. Hurtz
Computer Directions for Schools, 1983
P.O. Box 1136
 Livermore, CA 94550

Teaching Computer Programming to Kids and Other Beginners: A Teachers Manual
Royal Van Horn
Sterling Swift Publishing Co., Inc., 1982
7901 South IH-35
Austin, TX 78744

Teaching Word Processing in the Elementary School
John T. Riley and Judie L. Hurtz
Computer Directions for Schools, 1983
P.O. Box 1136
Livermore, CA 94550

Turtle Sourcebook
Jim Muller and others
11480 Sunset Hill Rd.
Reston, VA 22090

Using A Microcomputer in the Classroom
Gary G. Bitter and Ruth A. Camuse
11480 Sunset Hills Rd.
Reston, VA 22090
ASSOCIATIONS FOR EDUCATION AND COMPUTING

Association for Computers in Mathematics and Science Teaching
P.O. Box 4455
Austin, TX 78765
(512)258-8083

Association for the Development of Computer-Based Instructional Systems (ADCIS)
ADCIS Headquarters
409 Miller Hall
Western Washington University
Bellingham, WA 98225
(206)676-2860

Association for Educational Communications and Technology (AECT)
1126 16th Street NW
Washington, DC 20036
(202)833-4180

Association for Educational Data Systems (AEDS)
1201 16th Street NW
Washington, DC 20036
(202)822-7845
American Educational Research Association
1230 17th Street NW
Washington, DC 20036
(202)223-9845

Computer Education Group
North Staffordshire Polytechnic Computer Center
Blackheath Lane
Stafford ST18 OAD England

International Council for Computers in Education (ICCE)
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403
(503)686-4414

National Association of Elementary School Principals (NAESP)
1801 North Moore St.
Arlington, VA 22209
(703)528-6000

National Association for Secondary School Principals (NASSP)
1904 Association Drive
Reston, VA 22091
(703)860-0200

National Audio-Visual Association (NAVA)
3150 Spring Street
Fairfax, VA 22031
(703)273-7200

National Council of Social Studies (NCSS)
3615 Wisconsin Avenue NW
Washington, DC 20016
(202)966-7840

National Council of Teachers of English
1111 Kenyon Road
Urbana, IL 61801
(217)328-3870
National Council of Teachers of Mathematics (NCTM)
1906 Association Drive
Reston, VA 20091
(703)620-9840

National Science Teachers Association (NSTA)
1742 Connecticut Avenue NW
Washington, DC 20009
(202)328-5840

School Science and Mathematics Association
126 Life Sciences Building
Bowling Green State University
Bowling Green, OH 43403
(419)372-0151

Society for Applied Learning Technology
50 Culpepper Street
Warrenton, VA 22186
(703)347-0055
Conduit
P.O. Box 388
Iowa City, IA 52244
(319)355-5789

*Distributes and reviews software primarily focused on higher education, although some of it is appropriate for advanced high school science and math classes.*

Digest of Software Reviews
School and Home Courseware, Inc.
1341 Bulldog Lane, Suite C
Fresno, CA 93710

*Provides reviews of educational software.*

Microcomputer Education Applications Network (MEAN)
256 North Washington Street
Falls Church, VA 22046
(703)536-2310

*Provides information to help educators develop and sell software. The MEAN Brief, a quarterly newsletter, contains information on other software sources and industry news and provides a subscriber exchange.*
Minnesota Educational Computer Consortium (MECC)
2520 Broadway Drive
St. Paul, MN 55113
(612)376-1118

*Produces educational software and also publishes reviews of educational programs.*

Northwest Regional Educational Laboratory
300 S.W. Sixth Avenue
Portland, OR 97204
(503)248-6800

*Reviews software for educational applications.*

Softswap
c/o Ann Lathrop
San Mateo County Office of Education
333 Main Street
Redwood City, CA 94063
(415)363-5472

*This is a joint project of the Microcomputer Center of the San Mateo County Office of Education and Computer-Using Educators (CUE). Provides evaluations of software and refines programs that need work. An educator who contributes an original program on disk may request a swap for another program.*

Software Review
Microform Review, Inc.
520 Riverside Avenue
Westport, CT 06880
(203)226-6967

*Reviews programs for library and educational applications. Also contains articles on software concepts and evaluation. The review is published twice yearly.*
As you can see this is the exotic version of computer camps. Basically these camps try to demystify computers for adults who really need to know how to use computers for their jobs while allowing them a wonderful vacation at the same time. In most locations there are also mini clubs for children.
THE ORIGINAL COMPUTER CAMP, INC.
559 San Ysidro Road
Santa Barbara, CA 93108
(800)824-3349 (California)
(800)235-6965 (outside of California)
Camp Locations:
Santa Barbara, California
Steamboat Springs, Colorado
New Milford, Connecticut
The Poconos, Pennsylvania
Lake Tahoe, Nevada
Meridith, New Hampshire

Adult and family programs are available throughout the year.

CAMP MCALISTER
Greater New York YWCA
Huguenot, NY 12746
(212)564-1300 ext. 313

or

CAMP TALBOT
Greater New York YMCA
Huguenot, NY 12746
(212)564-1300 ext. 312

CHAMPLAIN COLLEGE COMPUTER CAMP
Champlain College
163 Willard St.
Burlington, VT 05403
(802)659-0800

COMPUTAR
Old Hardwick Road
Barre, MA 01005
(617)355-2164

MIDWEST COMPUTER CAMP
9392 Lafayette Rd.
Indianapolis, IN 46278
(317)297-2700

NATIONAL COMPUTER CAMPS
P.O. Box 585
Orange, CT 06477
(203)795-9667

Camp Locations:
Connecticut
Georgia
Missouri
Ohio
Oregon
SUN VALLEY COMPUTER CAMP  
P.O. 1450  
Hailey, ID 83333  
(208)788-2164  

THE TIC COMPUTER CAMPS  
Mount Vernon College  
Washington, D.C. 20007  
(703)241-5542  

YMCA OF THE BRONX  
2244 Westchester Avenue  
Bronx, NY 10462  
(212)931-2500  

RESOURCE BOOKS  

Parent’s Guide to Accredited Camps ($6.95)  
American Camping Association  
Bradford Wood  
Martinsville, Indiana 46151  

CAMP CONSULTING FIRMS  

Student Camps and Trips Advisors  
244 Bonad Road  
Chestnut Hill, MA 02107  
(617)469-0681  

Camp Advisory Services  
18 East 41st Street  
New York, NY 10017  
(212)696-0499
Each individual teacher should fill out a form designating requirements for his or her particular classroom situation, then a planning sheet for the entire school can be filled out using information from the individual forms.

Question 1: Why Do We Need Computers in the School Classroom?

Question 2: What Do You Want the Computer to Do?

- Prepackaged Programs
- Drill and practice
- Tutorials
- Simulations
- Educational games
- Classroom Administration and Tools for Teachers
- Generate tests and quizzes
- Keep track of grades and records
Attendance programs
Organize data
Write correspondence and records
Write programs for self and students
Create graphics and visual aids
Demonstrate
Generate IEP's
Prepare reports

Teaching Computer Literacy
Programming in BASIC
Ethics
History
Vocabulary

Teaching Programming
LOGO
PASCAL
COBOL
FORTRAN

Vocational Training
Accounting and bookkeeping
Computer science
Business skills/data entry, word processing, typing

Special Students
Teaching physically handicapped students
Teaching dyslexic students
Teaching slow learners

Question 3: What Hardware and Peripherals Will You Need to Meet Your Computer Use Objectives?

How many computers do you need?
How many hours per day/week/month will you need the computers? (Should you be filling out this form with another teacher who can share computers with your classroom?)
How many monitors?

- Monochrome
  - 40-column (number)
  - 80-column (number)

- Color
  - 40-column
  - 80-column

Keyboard—standard typewriter type/other (S/O)

How many printers are required?

- Dot-matrix (number)
- Letter-quality (number)
- Daisy wheel (number)
- Graphics printer—Color (number)

How much memory is needed?

- 64KB
- 128 KB
- 256 KB or more

Will you need diskette drives for your computers?

- One diskette drive
- Two diskette drives

Do you need portable computers? How many?

Will you need optional input devices?

- Joysticks (how many)
- Mice (how many)
- Graphics tablets (how many)
- Light pens (how many)

What programming languages will you need?

- BASIC
- LOGO
- PASCAL
- COBOL
- FORTRAN
- Other

How many modems will you need?

Question 4: What Will It Cost to Purchase Computer Equipment?

(Using the information from the previous question, determine what it will cost to purchase your computer system. It may be that once you price out all components for your computerized classroom you may find that you can get a package deal which will change the costs. The first time around, figure your expenses on individual components and make adjustments for package deals later. We suggest that you do this since a package may limit you somewhat on your choices and we want you to create an "ideal" shopping list first and then begin the compromise process.)


<table>
<thead>
<tr>
<th>Computerized Classroom Planning Sheet</th>
</tr>
</thead>
</table>

| #__________ | 64 KB computers × cost each = | ________ |
| #__________ | 128 KB computers × cost each = | ________ |
| #__________ | 256 KB computers × cost each = | ________ |
| #__________ | Additional disk drives × cost ea. | ________ |
| #__________ | Monochrome monitors 40-column × cost ea. | ________ |
| #__________ | Monochrome monitors 80-column × cost ea. | ________ |
| #__________ | Color monitors 40-column × cost each | ________ |
| #__________ | Color monitors 80-column × cost each | ________ |
| #__________ | Dot-matrix printers × cost each | ________ |
| #__________ | Daisy-wheel printers × cost each | ________ |
| #__________ | Graphics printers × cost each | ________ |
| #__________ | Letter-quality printers × cost each | ________ |
| #__________ | Joysticks × cost each | ________ |
| #__________ | Light pens × cost each | ________ |
| #__________ | Graphics tablets × cost each | ________ |
| #__________ | Mice × cost each | ________ |
| #__________ | Modems × cost each | ________ |

TOTAL COST OF SYSTEM $________

Question 5: Where Do You Find the Money to Buy Computers for Your School?

(If any category is a possibility, check and fill in amount you hope to receive. Describe how you might go about receiving such funds.)

1. Funding from local school districts. $________
2. Funding from the PTA. $________
3. Funding from local businesses. $________
4. Funding from computer manufacturers. $________
5. Funding from community organizations. $________
6. Private foundation grants $________
7. State Board of Education grants $________
8. Federal grants $________

Question 6: Where Will You Put the Computers?

(Describe where the computers will be installed and draw an illustration of that setting placing all components.)
Question 7: Who Will Teach the Teachers?

(Describe how you as well as other teachers can prepare yourselves to teach students about computers.)

Question 8: What Kinds of Software Will You Need?

(List here the kinds of software you will need immediately. This list will grow over time as you become more experienced with what is available and how much it costs. If you are familiar with specific programs that would be useful, list them; otherwise list the kind of program you need.)
active window: The currently usable window on the Macintosh screen. The frontmost window wherein any user's action will take place. If a window is active, that window's title bar will be highlighted.

address: A character or set of characters used to identify where in memory a particular datum is stored.

applications program: Programs that do specific tasks such as word processing or accounting, as opposed to educational programs, systems software, and games.

artificial intelligence: That branch of computer science concerned with designing computers to think like human beings. AI programs can learn from their own mistakes and modify themselves so they won't make the same mistake again.

ASCII (American Standard Code for Information Interchange): The code which is most commonly used for transmitting text between computers, or between a computer and its peripherals.

assembly language: A programming language used to talk to the computer. It is mnemonic, so that it can be easily understood and remembered by humans. However, it is not as easy to use as a high-level language like BASIC, which is even farther away from machine language and closer to human speech.

background: A computing task that is carried out by the computer processor while something different is seen on the computer screen. For example, the computer may be sending output to a printer while another document is being edited on-screen; the printing task is said to be the "background task."

backup: A copy of a diskette that is made so that data will not be lost if the original diskette is damaged. Usually, the backup copy will be used and the original stored in a safe place.

BASIC: Beginner's All-Purpose Symbolic Instruction Code. The programming language which is permanently imbedded in the PC's ROM. It is the language used in most personal computers.

batch file: A batch file is a file containing commands that the DOS system acts on one at a time.
baud rate: The number of bits that can be transmitted or received by a modem.

benchmark: A standard unit of testing or a standardized testing program used to measure a computer's performance.

binary: Refers to the numbering system used by a computer that is made up of only zeros and ones, providing a selection or condition of only two possible values or states. The individual digits of this numbering system are called bits and it is these bits that are the basic units of data that a computer processes.

binary numbers: Composed entirely of zeros and ones, these are the numbers computers understand. In this system all values are expressed in powers of two.

BIOS: Basic Input/Output System. A computer's built-in operating code.

bit: Short for "binary digit," this is the smallest possible unit of information. A bit can say "yes" or "no" or "on" or "off."

blocks: A set of things, such as characters, words, or digits, that are handled as a unit. Usually data is processed one block at a time.

boot: The process of starting up the computer by loading the operating system into it.

bug: A mistake or malfunction that prevents a software program from running properly.

bus: A conductor in the system unit used for transmitting signals or power between the CPU and memory, I/O boards, and other devices in a computer.

byte: A string of binary characters composed of 8 bits. This is the amount of information required to define one character.

CAI: Computer-aided instruction—using computers to teach.

click: Represents the action of pressing the Macintosh mouse pointer button. If the pointer is currently over a screen feature that can be activated, this action will select that feature.

clipboard: The storage place for text that was last designated to be cut or copied.

close: To end activity in a particular window. Closing a window removes it from the screen and leaves only the icon that represents the related program.

close box: The representation of a small box found at the left edge of an active window's title bar. Clicking the mouse pointer over the close box closes that window.

COBOL: An "English-like" programming language that is widely used in business applications.

command files: Files used by the operating system for basic functions such as file management.

configuration: An arrangement of connected hardware devices, which may vary according to a user's needs.

controller board: See printed circuit board.

CPU: Central Processing Unit. This is the part of the computer that actually does the computing.

CRT display: Cathode-ray tube display.

cursor: An indicator on the display screen that marks the position where the next data will be entered, or that indicates an item in a menu the user wants to select.

cycling: The repetition of a function, such as a character displayed on the screen, an editing function, or a diagnostic test.

data: Formalized representations of facts, instructions, or concepts that allow communication, interpretation, or processing by humans and computers.

database: A set of data essential to a particular data-processing system or enterprise.

data bus: Used to communicate data internally and externally, to and from the processing unit, storage, and peripheral devices.

data processing: A systematic manipulation of data, such as sorting, computing, merging, etc.

debug: To go through software locating and eliminating mistakes that prevent it from running.

dedicated: Reserved for one particular use or purpose.

default: A preset alternative that will determine your option, if no other option is provided to override that which is built into the program.

density: A term used to describe the amount of data a diskette can store; a double-density diskette can store twice as much data as a single-density diskette.

diagnostic diskette: A diskette containing special test programs used to isolate errors in the system.

dialog box: A box, which appears on the Macintosh, that requests a response from the user.
diskette: (Also called a flexible or floppy disk)—A magnetic disk for recording and storing data files or programs. It is similar in principle to magnetic recording tape used to record sound.

diskette drive: The hardware device that reads from or writes to the diskette. The diskette must be inserted in the diskette drive in order for its information to be accessed or for new information to be added to it.

display monitor: A cathode-ray tube (CRT), such as a television screen, that provides a visual display of the information you are working with.

documentation: Instructions, training manuals, reference materials, and all the other written information about your computer’s hardware and software.

DOS: Disk operating system.

double clicking: Pressing the Macintosh mouse pointer button twice. This action can be used to both select and activate a program.

dragging: Pressing and holding down the Macintosh mouse button while sliding the pointer across the tabletop.

download or downline load: To transfer data from a computer in another location to a local computer, usually via telephone lines. (Compare with upload.)

execute: To perform an operation. Computers are said to execute instructions.

external commands: DOS commands that are not loaded into the system unit’s memory, but only retained on the DOS diskette, so that the system has to look them up on the diskette when it needs them.

file: 1. A document file: A collection of information, such as a document or a record, compiled by the user and stored on a diskette.  
2. A program file: a collection of data used to control the computer. (These files also may be created by the user.)

file codes: Units of information about files, required by some commands to specify exactly what the command is to do.

filename: A name assigned by a user. The system recognizes the filename so the user can refer to it when specifying which file is to be acted on.

firmware: Part of a computer’s operating system that occupies a gray area somewhere between hardware and software. It is coded like software, but is a permanent part of the hardware system, stored in the read-only memory (ROM).

format: A procedure that prepares a diskette to be compatible with the operating system of the computer being used.

function keys: Special keyboard keys that can be programmed to perform an entire operation with one keystroke.

hardcopy: Computer term for printer output; in other words, printed paper.

hard disk: A magnetic storage device used for high-volume data storage.

hardware: The physical devices comprising a computer system, such as the system unit, display monitor, keyboard, and printer.

help screens: Helpful messages about how to use a computer or a computer program that is built into a program. Often one key on the keyboard is designated as the “help key”; pressing that key makes the help screen messages appear.

hidden files: Command files used by the operating system. These files are not normally displayed as part of the operating system directory.

initialize: A procedure that prepares a diskette to be compatible with the operating system of the computer being used. (Same as format.)

input/output (I/O): The reception and transmission of data; a device that both receives and transmits data. For example, a keyboard is an input device. A printer is an output device. A diskette drive is an input/output device since it can be written to or read from.

insert mode: A state of text entry in which characters entered shift all subsequent characters to the right, instead of overwriting them.

insertion point: The place in the on-screen text where the next text entered will be inserted. When using a program on the Macintosh, the insertion point is selected by clicking, and is represented by a blinking vertical line.

integrated programs: Multiple computer programs accessed through one central program. To the user, these programs appear to be one, multipurpose program.

interface: A shared boundary that links together two separate components. A hardware interface connects two different hardware components. (Also see user interface.)
internal commands: DOS commands that are loaded into the system unit's memory when DOS is loaded, so that they are still available even if the DOS diskette is removed.

internal files: Operating system files that are loaded into random access memory (RAM) whenever the operating system is loaded.

I/O Bus: The communication lines inside the computer along which all the input and output activity are transmitted.

joystick: A device, often used in video games, that can be pointed in any direction to move the cursor around the screen.

light pen: A device used to draw designs or enter data on a special CRT screen that has been sensitized to respond to it.

load: To enter information, such as an operating system or other program, into the computer, frequently from a diskette or disk.

LOGO: A high-level programming language developed especially for children. The term comes from the Greek word logos, which means both "word" and "idea," and is meant to emphasize the connection between linguistic and mathematic ways of thinking.

mainframe: Large-scale computers, usually used to provide computing services to a large number of users.

megabyte: One million bytes.

memory: The area of the computer where data is copied, stored, and later retrieved.

menu: A list of choices that appears on the screen and guides the user in choosing which operation to request.

menu bar: The strip across the top of the Macintosh screen that designates the names of the available menus.

microcomputer: A small sized computer. Usually considered to be any computer built around a microprocessor.

microprocessor: The part of a microcomputer that does the actual computing. The circuitry required is squeezed onto a single chip.

microsecond: A millionth of a second.

millisecond: A thousandth of a second.

mode: A particular method or state of operation.

modem: A device that allows one computer to talk to another, or terminals to talk to computers, over phone lines.

modem board: A board or card that plugs into the computer's motherboard and functions as a modem.

monitor: See display monitor.

monochrome: Single color, such as black on white or white on black. In this case of the IBM, the monochrome screen is green on black.

motherboard: A board in the system unit where cards for peripherals can be plugged in.

mouse: The pointing device that can be moved across a tabletop to move a corresponding pointer across the Macintosh screen.

multitasking: The ability of a computer or a computer program to carry out more than one task at a time. For example, a word processing program may allow editing and printing at the same time.

multiuser: A computer processing system that can be used by more than one user at a time. Multiuser hardware and software must utilize some method of timesharing during which the processor keeps track of each user's input, yet allows each user periodic, individual access to the processor.

nanosecond: One billionth of a second.

network: A system in which several computers or computer terminals are connected to each other.

number crunching: Tedious, repetitive processing of numerical data.

operating system: The underlying program that tells the computer what to do. This may vary from one computer to the next. It is what determines whether one computer is compatible with another.

operator interface: The point of interaction between a computer user and computer hardware and programs. The focus is on the ease or difficulty of contact at the boundary between user and computer. (Same as user interface.)

output: Data that the computer puts out after it has been processed.

parallel printer interface: The device or method by which a computer sends simultaneous bits of data to a printer. (Contrast with serial printer interface.)

parallel transmission: Concurrent or simultaneous transmission of data bits across a circuit. (Contrast with serial transmission.)

peripherals: Separate hardware devices, such as printers or modems, that are attached to and under the control of a computer.

picosecond: A trillionth of a second.
pixel: The smallest element on a display screen that can be assigned color or intensity. The more pixels there are, the higher the resolution will be.

printed circuit board: A unit consisting of electronic parts on a board. Electronic impulses from these boards, directed by programs, make the computer work.

program: A set of instructions that tells the computer what to do. It is also called software and can come in a variety of forms, including floppy disks, cassettes, and hard disks.

programming language: A program that is used to create other programs. Each instrument in the language corresponds to machine code instructions.

prompt: A message that appears on the screen to remind the user to press a key or enter some data.

protocol: Rules and standards by which data is transferred. Required whenever two computers are communicating.

RAM: Random access memory. This is memory you can both write data into and read data out of.

read-only: Usually, a program that the user can read from but cannot change; for example, the system operating software.

read/write head: An electromagnetic device, similar to the head in a tape recorder, that can read from or write to magnetic storage devices such as diskettes.

ROM: Read-only memory. This is memory that tells the computer what to do, but is built permanently into the system so that you cannot change it. You can read it, but you can’t write on it.

scroll: To move an entire screenful of text up or down on the computer’s screen.

sector: One section of a concentric track on a diskette. In order to keep track of stored data, computers label storage addresses sector by sector.

serial transmission: Sequential transmission of data bits across a circuit. Computer communications over telephone lines are commonly sent and received serially. (Contrast with parallel transmission.)

shell: Methods of giving a computer user access to complicated computing techniques without the user having to learn the details of the computing system. Commonly, shells use such ease-of-use techniques as menus and help screens to make using the programs easier. The sum of these techniques for any program constitutes the shell.

shift-clicking: Pressing the Macintosh’s mouse pointer button while holding down the shift key.

size box: The representation of a small box at the lower right of an active window used to alter the size of the window.

software: The instructions, or programs, that are used to direct the operations of a computer.

source disk: The disk from which data is being transferred or copied.

storage address: A particular area in storage. The computer maintains a system of addressing each discrete area in order to be able to quickly locate stored data.

structured programming: Methods of programming that group classes of programming activity into predefined sections.

subroutine: A part of a program (routine) that may be called into operation under certain conditions.

system board: See motherboard.

system unit: The central computer of the IBM PC computer system, containing the main memory, and the printed circuit boards for attaching the other hardware devices in the system.

target disk: The disk to which data is being transferred or copied.

temporary memory: Refers to memory storage within the system unit that is erased every time the unit is turned off. Files the user wants to keep should be stored on a diskette or cassette recorder cartridge.

title bar: The representation of a horizontal bar along the top of a window that shows the name of the program represented by that window.

toggle: Any device, such as switch, that has two states (on and off). Pressing the switch once initiates one state; pressing it again initiates the other.

track: One of the concentric grooves on a diskette. Computers use a method of addressing tracks and sectors to keep track of stored data.

upload: To transfer data from a local computer to a computer in another location, usually via telephone lines. (Contrast with download.)

user interface: The point of interaction between a computer user and computer hardware and programs. The focus is on the ease or difficulty of contact at the boundary between user and computer. (Same as operator interface.)
user memory: Another term for random access memory.

wild card: A character that can be used to take the place of one or more unknown characters in a filename.

window: A display area on the Macintosh screen. Many different windows can be displayed on the screen at the same time and different programs can be accessed through different windows.

write-protect: To protect a disk or diskette from being changed (written on). In the case of the DOS software, for example, the diskette is write-protected so that the operating system can't be altered by the user or the computer. See also read-only.
INDEX

A

Administrative programs, 116–17
Alarm clock, 51, 60
Analog board, 39
Apple Corporation, 10
AppleDOS, 46
Apple II, 8
Atari, 5, 10–11
AT&T, 6, 77
Authoring languages, 118, 124

B

Background processing, 52
Back panel, 38

BASIC, 64–73
commands:
CLS, 69
DATA, 69
DEL, 69
DIM, 69
DRAW, 70
EDIT, 70
END, 70
FILES, 70
FOR . . . NEXT, 70
GOSUB . . . RETURN, 70
GOTO, 70
IF . . . THEN, 71
INPUT, 71
LET, 71
LIST, 71
Index

LLIST, 71
LOAD, 71
LOCATE, 71
LPRINT, 72
NAME, 72
NEW, 72
PLAY, 72
PRINT, 72
READ, 72
REM, 72
RENUM, 72
RUN, 72
SAVE, 72
SOUND, 72
SYSTEM, 72

BIOS, 25 (def)
Bit, 7 (def)
Bulletin boards, 85–86
Business software, 14

C

Calculator, 50, 58
Click, 31, 56
Close box, 55
COBOL, 123
College exam preparation, 102, 112–13
Communications software, 79
Computer games, 14
Consumer database, 86–87
Control panel, 61
CP/M, 45
CPU, 9, 16–19

D

Database, 84 (def)

calculator, 50, 58
edcational, 88–89
Desk accessories, 58
Desktop, 49, 53
Digital Research, Inc., 45
Diskette drive, 36–38
Diskettes, 36–38
format, 36
initialize, 36
sector, 36
storage address, 36
track, 36
write protect, 37
Double click, 31, 56
Drag, 32, 57
Dragging windows, 57
Drill-and-practice programs, 111–13

E

Education:
associations, 182–84
home, 89–104
college entrance exams, 102
computer camps, 113–14, 187–89
computer literacy, 98
encouraging math skills, 99
first encounter, 91–92
home curriculum, 93
modems, 102–3
preschoolers, 95–97
programming, 99–100
software review centers, 185–86
timesharing, 92
typing, 99
user's groups, 97–98
word processing, 101–2
school, 105–43
authoring languages, 118, 124
business classroom, 127
classroom administration, 116-17
classroom planning sheet, 190-93
COBOL, 123
college entrance exams, 112-13
computer literacy, 106-7, 118-19
cost of computers, 126, 131-32
databases, 130-31
drill-and-practice programs, 111-13
educational computer magazines, 168-74
educational computing books, 175-81
educational games, 115-16
financing computers, 133-35
FORTRAN, 123
LOGO, 121-22
Pascal, 123
programming, 120-24
recordkeeping, 118
selecting educational software, 137, 143
simulations, 114-15
special students, 125-26
teaching aid, 110-11
teaching teachers, 136-37
testing, 112
tutorials, 113-14
vocational training, 124-25
word processing, 117-18
Educational games, 115-116
Educational software, 15, 144-67
8-bit CPU, 9, 20
Encouraging math skills, 99

F

Financing school computers, 133-35

Finder, 40-42, 47, 52
Firmware, 25, (def)
Fixed disk, 24
Format, 36

H

Hard disks, 24
Hardware, 4, (def)

I

IBM PC, 4-6
IBM PC-AT, 8
IBM PCjr, 10
IBM standard, 4-7, 48
Icons, 56
Incompatible BASIC programs, 65
Intel 8088 CPU, 9, 16-17, 19

K

Keyboard, 28-29, 32
command key, 34
enter key, 33
option key, 33-34
numeric keypad, 35
special keys, 33
typewriter keys, 32-33
Key caps, 61

M

MacEdge, 146-47
MacPaint, 3, 13
MacProject, 148
MacTerm, 15
MacWrite, 3, 13
Memory, 21–24
  RAM, 20, 22–23
  ROM, 22, 24
Menu, 46
Modems, 140–61
  abort command, 80
  analog signals, 76
  automatic answer, 81
  baud rates, 77–78
  Bell 103, 77
  Bell 212A, 77
  busy mode, 81
  connect signals LEDs, 80
  Dialing Directory, 81
  digital signals, 76
  downline load, 82
  full duplex, 80
  half duplex, 80
  help command files, 80
  history, 81
  line monitor speaker, 79
  manual dial, 81
  number linking, 81
  protocal detect, 81
  protocols, 76–77
  redial last number, 81
  serial transmissions, 76
  speed, 77
  standards, 75–76
  V.22 protocol, 77
Modulator/demodulator, 76
MOS 6502 CPU, 8, 18–19
Motherboard, 39
Motorola 68000 CPU, 8–9, 16–17, 19, 21, 39
Mouse, 4, 7–8, 28–32
Multitasking, 50
Multiplan, 149

N
  National Semiconductor, 19
  NCR, 19

O
  Operator interface, 42

P
  PALs, 39
  Parallel data transfer, 76
  Pointing, 56
    BASIC, 64–73
    COBOL, 123
    FORTRAN, 123
    Logo, 121–22
    Pascal, 123
  Program menus, 56
  Pull-down menu, 55
  Puzzle, 61

R
  Radio Shack, 4, 6, 10, 48
  Random access memory (RAM), 20, (def) 22, 24
  Read only memory (ROM), 20, (def) 22, 24

S
  Sargon III, 148
  Scroll arrows, 55
  Shell, 46
Shift click, 32
16-bit CPU, 9, 20
Size box, 55
Software, 2 (def), 4
  accounting, 15
  education, 15
  business, 14
  innovations in, 3
  math, 15
  scientific, 15
  under development, 152–67
Sperry Corporation, 6

T

Texas Instruments, 4
Think Tank, 150–51
32-bit CPU, 9–20
Title bars, 54

Trash feature, 57
TRSDOS, 46

W

Windows, 52–54
Word, 149
Write-protect, 37

X

Xerox, 30, 41, 47

Z

Zilog, 19
This book is designed to inform parents and educators about the Apple Macintosh as a practical choice for educational applications at home and in the classroom. Everett Murdock and Susan Sudbury provide technical background material to help you make comparisons to other personal computers on the market. They describe the capabilities of the Macintosh in detail, including Apple's remarkable user software.

Among its highlights, the book:

- describes how parents can introduce preschoolers to the Apple Macintosh and also describes what programs they can use.
- shows how schools can fund their computer purchases.
- suggests how to make the Macintosh a member of the family.
- discusses the new features of the Macintosh and how you can best use them.
- takes you behind the user friendliness of the Macintosh to introduce you to a powerful computer.

Of special interest:

- Illustrations and diagrams.
- Charts and tables.
- Background material on computers and on educational computing.
- Glossary of computer terms.
- Appendices with resources for educational computing.

PRENTICE-HALL, INC., Englewood Cliffs, N.J. 07632