

IBM PCjr

Buyer and User Guide



Larry Joel Goldstein

IBM PCjr

Buyer and User Guide

Larry Joel Goldstein, Ph.D.

Robert J. Brady Co., Bowie, MD 20715
A Prentice-Hall Publishing and Communications Company

NOTE TO AUTHORS

Do you have a manuscript or a software program related to personal computers? Do you have an idea for developing such a project? If so, we would like to hear from you. The Brady Co. produces a complete range of books and applications software for the personal computer market. We invite you to write to David Culverwell, Editor-in-Chief, Robert J. Brady Co., Bowie, MD 20715.

Editor-in-Chief: David Culverwell
Production Editor/Text Design: Paula Huber
Art Director/Cover Design: Don Sellers
Assistant Art Director: Bernard Vervin

Typesetter: Alexander Typesetting, Indianapolis, IN
Printer: R.R. Donnelley and Sons Company, Harrisonburg, VA
Typefaces: Times Roman (text), Bookman (display), OCR-B (programs)

IBM PCjr Buyer and User Guide

Copyright © 1984 by Robert J. Brady Company.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage and retrieval system, without permission in writing from the publisher. For information, address Robert J. Brady Co., Bowie, Maryland 20715.

Library of Congress Cataloging in Publication Data

Goldstein, Larry Joel.
IBM PCjr.

Includes index.

1. IBM PCjr—Programming.

I. Title. II. Title. III. I.B.M. P.C.jr.

QA76.8.I2593G65 1983 001.64 83-25868

ISBN 0-89303-546-7

Prentice-Hall International, Inc., London
Prentice-Hall Canada, Inc., Scarborough, Ontario
Prentice-Hall of Australia, Pty., Ltd., Sydney
Prentice-Hall of India Private Limited, New Delhi
Prentice-Hall of Japan, Inc., Tokyo
Prentice-Hall of Southeast Asia Pte. Ltd., Singapore
Whitehall Books, Limited, Petone, New Zealand
Editora Prentice-Hall Do Brasil LTDA., Rio de Janeiro

Printed in the United States of America

84 85 86 87 88 89 90 91 92 93 94 10 9 8 7 6 5 4 3 2 1

CONTENTS

1. A First Look at Computers

1.1 So You Want to Buy a Home Computer	1
1.2 What Is a Computer?	2
1.3 A Grand Tour of the PCjr	5

2. What Can You Do With the PCjr?

2.1 Why Buy a Personal Computer?	7
2.2 Entertainment	8
2.3 Home Finance	8
2.4 Word Processing	14
2.5 Education	16
2.6 Do-It-Yourself Programming	16
2.7 Communications	17
2.8 Graphics and Sound on the PCjr	17
2.9 Conclusion	21

3. A Close Look at the PCjr's Hardware

3.1 The Keyboard	23
3.2 System Unit	24
3.3 Cassette and Diskette Storage	31
3.4 Displays	32
3.5 Printers	33
3.6 Communications	34
3.7 The Serial Port	36

4. Software & Publications for the PCjr

4.1 BASIC on the PCjr	37
4.2 Unique Features of PC BASIC	38
4.3 Learning PC BASIC	46
4.4 Summary of PCjr BASIC	47
4.5 What Is DOS?	56
4.6 IBM Software and Publications	57
4.7 IBM SAMPLER and Tutorials	64

5. Typical System Configurations

5.1 A Starter System	65
5.2 An Upgraded Starter System	66

5.3 A Minimal Diskette System	67
5.4 A Deluxe System	69
5.5 What Will It All Cost?	70

6. Other Members of the IBM Family

6.1 The IBM PC and IBM PC/XT	77
6.2 Software Compatibility	79

7. PCjr System Planning Aids

7.1 Planning a PCjr System	81
7.2 System Component Summary and Price List	81

USER GUIDE TO DOS 2.1

8. Using DOS for the First Time

8.1 On Diskettes and Diskette Drives	85
8.2 Starting Your PCjr	88
8.3 More About Diskette Drives	91
8.4 Backing Up Your DOS Diskette	92
8.5 The Keyboard	94

9. An Introduction to DOS

9.1 Files and File Names	99
9.2 File Specifications	102
9.3 Executing Commands and Programs	104
9.4 The COPY Command	106
9.5 COPYing and FORMATting Diskettes	109
9.6 Other DOS Internal Commands	113
9.7 Other DOS External Commands	114
9.8 Creating Your Own DOS Commands—Batch Files	115

10. More About DOS 2.10

10.1 Paths, Directories and Subdirectories	121
10.2 More DOS Commands	128
10.3 Enhanced Batch File Capability	137
10.4 Redirecting Input and Output	143
10.5 DOS 2.1 Command Summary	144

Appendix: Hardware Specifications	147
--	-----

Answers to Exercises	152
-----------------------------	-----

Index	153
--------------	-----

PREFACE

I have found that the personal computing revolution has taken most people by surprise and has found them completely uninformed (at worst) or confused (at best). Most people accept (sometimes reluctantly) that the personal computer occupies an important place in our emerging "high tech" society, and that it is important for everyone, old and young alike, to become "computer literate." And one of the simplest, and most enjoyable routes to computer literacy is owning a home computer. However, *buying* a home computer is no simple task. You must choose from a bewildering array of computers and computer stores. And that's only the beginning! Once you decide on a particular computer, you must choose from a veritable smorgasbord of options to configure your system. This book is designed to take some of the pain and anxiety out of this process.

If you are thinking about purchasing a home computer system, you will certainly want to look at IBM's PC Junior. If industry observers are to be believed, the PCjr will rapidly set a standard for the field of personal computing. However, the PCjr is sold on an "a la carte" basis. A PCjr system is composed of a number of separate components from which you must choose in order to configure a working system. This book walks you through the system selection and purchase process, from the time you begin to think about buying a system, to the time you come home and set up your PCjr. My goal is to answer your questions about the system and to help you arrive at a configuration that is right for your needs.

First, we discuss the possible applications of the PCjr. Then we'll talk about the various system components, how they may be integrated into a system, and the various hardware and software options.

Personal computing is interesting, educational, and fun. I hope that this book smoothes your entrance into this exciting field.

Larry Joel Goldstein
Silver Spring, Maryland

Limits of Liability and Disclaimer of Warranty

The author and publisher of this book have used their best efforts in preparing this book and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regards to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

IBM, IBM PCjr, IBM PC, and IBM PC/XT are registered trademarks of International Business Machines Corp.

1

A FIRST LOOK AT COMPUTERS

1.1 So You Want to Buy a Home Computer

Are you thinking about buying a home computer? That's great! Welcome to the computer revolution! Are you considering IBM's PC Junior? If so, then this book is meant for you.

Purchasing a home computer can be a complicated task. First, you must decide what you want your system to do for you. Next, you must design your system from the available components. (Designing a computer system is similar to designing a stereo system from components.) Some of the components are pieces of equipment (hardware) and some are programs you plan to use (software). Finally, you must purchase your system, set it up, and learn to use it. We've attempted to assemble all of the information you need to make informed purchase decisions. Also, Chapters 8, 9 and 10 are a brief User Guide designed to teach you something about using the machine while you are still considering purchase or while you are awaiting delivery.

You won't need any background in computers. In the rest of this chapter, we'll introduce you to computers generally and to the PCjr system in particular.

In Chapter 2, we'll discuss a very essential question: Why do you want a home computer system? In that chapter, we'll discuss the various things you can do with a home computer. We'll cover some of your interests, I'm sure.

In Chapter 3, we discuss in detail the various hardware components of the PCjr system.

Chapter 4 is devoted to software. First we discuss BASIC (a computer language) on the PCjr. Then we give a brief description of the Disk Operating System (DOS). Finally, we give a survey of the IBM software offerings for the PCjr.

In Chapter 5, we describe four possible system configurations and determine the cost of the hardware.

Chapter 6 is devoted to a discussion of the similarities and differences between the PCjr and the two other IBM personal computers, the PC and the PC/XT. We also discuss the compatibility of software among the three systems.

In Chapter 7, we let you plan your own system. We provide a list of all components, with serial numbers and prices.

Chapters 8, 9 and 10 constitute a brief User's Guide for the PCjr. We discuss the keyboard, how to use the computer for the first time, and the fundamentals of the Disk Operating System.

1.2 What is a Computer?

At the heart of every computer is a **central processing unit** (or **CPU**) that performs the commands you specify. This unit carries out arithmetic, makes logical decisions, and so forth. In essence, the CPU is the "brain" of the computer. The **memory** of a computer allows it to "remember" numbers, words, and paragraphs, as well as the list of commands you wish the computer to perform. The **input unit** allows you to send information to the computer; the **output unit** allows the computer to send information to you. The relationship of these four basic components of a computer is shown in Figure 1-1.

In an IBM PCjr, the CPU is contained in a tiny electronic chip, called an **8088 microprocessor**. As a computer novice, you don't need to know about the electronics of the CPU. For now, view the CPU as a magic device somewhere inside the case of your computer and don't give it another thought!

The main input device of the PCjr is the computer keyboard. We will discuss the special features of the keyboard in Section 8.5. For now, think of the keyboard as a typewriter. By typing symbols on the keyboard, you are inputting them to the computer.

The PCjr has a number of output devices. The most basic is the "TV screen" (sometimes called the **video monitor** or **video display**). You may also use a printer to provide output on paper. In computer jargon, printed output is called **hard copy**.

There are four types of memory in a PCjr: **ROM**, **RAM**, **cassette**, and **diskette**. Each of these types of memory has its advantages and disadvantages. Microcomputers attempt to make memory as versatile as possible by using several kinds of memory, thereby allowing them to take advantage of the good features of each.

ROM

ROM stands for "Read Only Memory." The computer can read ROM but cannot write anything in it. ROM is reserved for certain very important programs necessary to the operation of the machine. These programs are re-

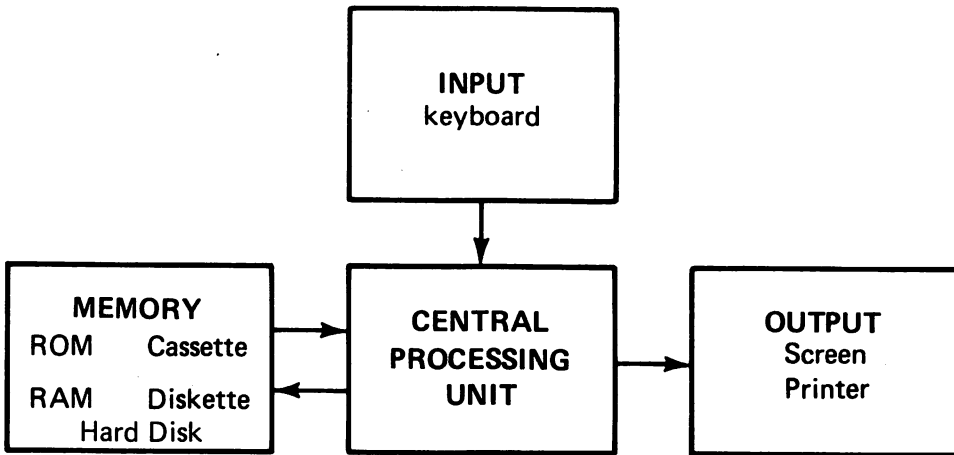


Figure 1-1. The main components of a computer.

corded in ROM at the factory and you may not change them. The PCjr has some ROM in its system unit. In addition, you may purchase ROMs containing various games or an enhanced version of BASIC. These ROMs may be plugged into one of the two cartridge slots on the PCjr.

RAM

RAM stands for “Random Access Memory.” This is the memory that you can read and write. If you type characters on the keyboard, they are then stored in RAM. Similarly, results of calculations are kept in RAM awaiting output to you. As we shall see, RAM even holds the instructions that perform the calculations!

There is an extremely important feature of RAM that you should remember:

If the computer is turned off, then RAM is erased.

Therefore, RAM may not be used to store data in permanent form. Nevertheless, it is used as the computer’s main working storage because of its great speed. (It takes about a millionth of a second to store or retrieve a piece of data from RAM.)

The size of RAM is measured in **bytes**. Essentially, a byte is a single character (such as “A” or “!”). You will often hear statements such as: “The PCjr comes with 64K of RAM.” The abbreviation “K” stands for the number 1,024. And 64K stands for 64 times 1,024 or 65,536 bytes.

To make permanent copies of programs and data, you can use either a cassette recorder or a diskette drive.

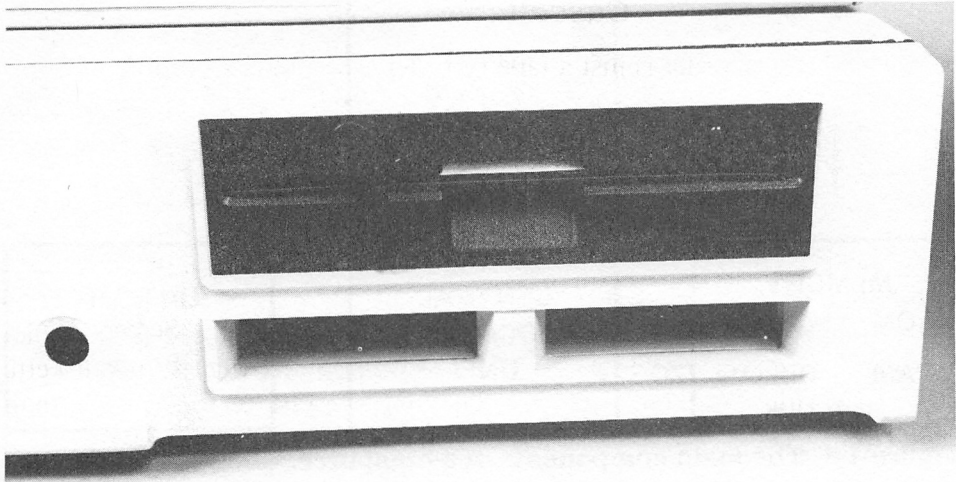


Figure 1-2. A diskette drive.

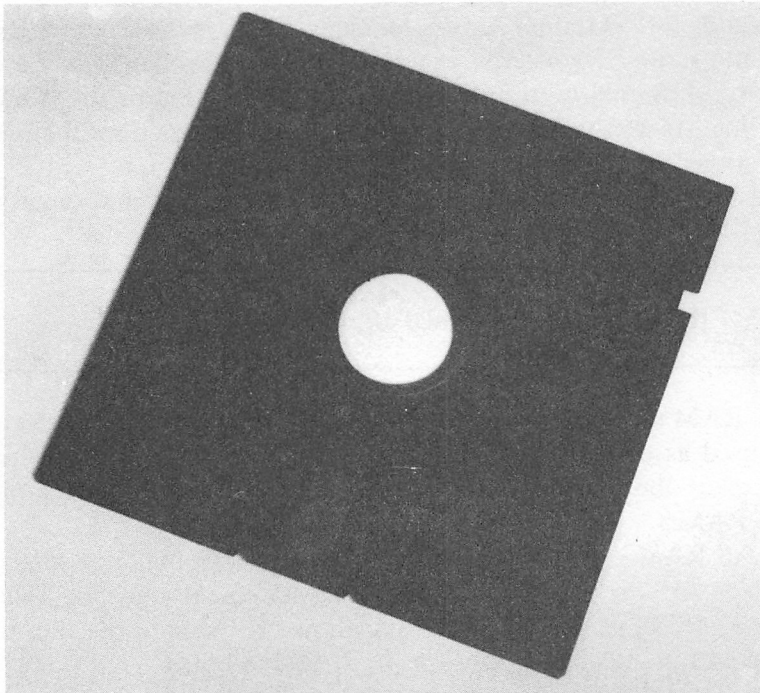


Figure 1-3. A floppy diskette.

Cassette Recorder

The cassette recorder is just a tape recorder that allows recording of information in a form the computer can understand. The recording tape is the same type you use for musical recordings.

Diskette Drives

A diskette drive (Fig. 1-2) records information on flexible diskettes that resemble phonograph records. The diskettes are often called “floppy disks,” and they can store several hundred thousand characters each! (A double-spaced typed page contains about 3,000 characters.) (See Fig. 1-3.) A diskette file can provide access to information in much less time, on the average, than a cassette recorder. On the other hand, diskette drives are more costly.

1.3 A Grand Tour of the PCjr

Now that we know the elements of computer systems in general, let's take our first look at IBM's PCjr.



Figure 1-4. The PCjr basic system.



Figure 1-5. An enhanced PCjr system.

The system comes in two versions: basic and enhanced. Let's describe the basic version, shown in Figure 1-4.

Note that there are three component parts to the BASIC system. The **keyboard** is the unit that looks like a typewriter. The **system unit** is the large box. The system unit contains the 8088 microprocessor and most of the electronics of the PCjr. Inside the system unit is 64k of RAM. The system unit has two cartridge slots into which you can insert program ROMs.

The other box is the transformer, which converts household electric current to a level the PCjr can use.

The enhanced system is shown in Figure 1-5. Note that this system has a diskette drive in the system unit. (The diskette drive is the long, rectangular door at the bottom of the unit.) In addition, the enhanced system has 128k of RAM and a display adapter capable of displaying an 80 character wide line.

We will describe the various PCjr components in greater detail in Chapter 3. But first, let's discuss the possible uses of a PCjr.

2

WHAT CAN YOU DO WITH THE PCjr?

2.1 Why Buy a Personal Computer?

In the past few years, the personal computer industry has received an incredible amount of publicity. As a result, you may have this vague feeling that you ought to go out and buy a computer, but you're not sure what you would do with one. This chapter is designed to help you remove that uncertainty.

Personal computing can literally affect your life. It can:

Improve your efficiency at work and at home.

Open up new career opportunities for you.

Assist in educational programs for all members of your family.

Expand your recreational possibilities.

In order to get the most out of a personal computer, you should ask yourself these two questions:

1. How can a personal computer help me?
2. Which personal computer system is up to the task?

If you buy a personal computer without having some clear uses for it, your computer will very quickly end up in the closet. On the other hand, if you have one or two realistic applications in mind, you will discover many more in your first few months of use.

In this chapter, we'll look at some of the applications of the IBM PCjr. As you read, ask yourself: Is this something I would want to do with my computer? The applications you specify will help you decide if you should buy an IBM PCjr and, if so, the equipment you'll need.

We couldn't possibly discuss *all* the applications of the PCjr. Instead, we have isolated a few of the most important, namely:

Entertainment
Home Finances
Word Processing
Education
Data Storage
Do-It-Yourself Programming
Graphics and Sound

We will give a reasonably detailed discussion of each of these applications.

2.2 Entertainment

In the past five years, a video game craze has swept the world. During this time, video games have progressed from relatively simple table tennis games to elaborate, multi-colored, arcade-style games, complete with music and sound effects. Hundreds of millions of quarters have been fed to arcade game machines and an equal amount to home video systems.

The PCjr can function as a very sophisticated home video system. Its multi-color graphics are comparable in quality to existing video systems. And its three-voice music synthesizer chip allows musical accompaniment in harmony, no less!

Games for the PCjr can be purchased on three different media: cartridge, cassette, and diskette. For the last two, you will need to equip your PCjr system with a cassette recorder or a diskette drive, respectively. However, all PCjr's are equipped to use cartridge games.

IBM has announced an initial offering of arcade-style cartridge games. In addition, there are a number of diskette-based arcade-style and strategy games that already exist for the PC and PC/XT, and that will run on the PCjr.

2.3 Home Finance

There are many opportunities to use a personal computer in the area of home finance.

Banking and Budget Records

You can keep all of your banking and budget records on the PCjr. You might start with your checking account. By recording all your checks and deposits, your computer could balance your checkbook each month. In all honesty, however, if this is the only financial application you plan for your computer, you probably shouldn't bother.

As you'll find, it's a time-consuming job to enter all your checks and deposits. And, if the system is to work, you must constantly update your records. If the only benefit you plan to derive from your record-keeping is a balanced checkbook, you could accomplish the same goal more quickly with a pencil and a scratch pad. However, balancing the checkbook is just a start.

As you enter your expenses, you could divide them into budget categories (rent, utilities, food, entertainment, etc.) Each month, you could have the computer prepare budget totals, by spending category. Also, you could prepare year-to-date summaries, comparisons with last year, and so forth.

You can buy PCjr programs that will easily help you to keep all of the above records and generate the statistics.

Tax Planning and Tax Returns

You could use your computerized budget to keep track of tax information. At the end of the year, you would have all your income and deductions in a neat and orderly form. No more scrounging through piles of receipts trying to make sense of a whole year's worth of forgotten transactions.

There are even programs that will compute your tax return for you. You feed in the raw data on income and deductions, and they will calculate the various numbers to go on the form. Some of the programs will even print out the data onto the tax forms.

Investing

The PCjr can be a great help in managing investments and making investment decisions.

First, you can use the PCjr to keep track of your investments: stock trade dates, buy and sell prices, capital gains and losses, dividends received, coupons to clip, and so forth.

Second, you may use the PCjr along with available software to track individual stocks. You can make tables of stock prices on a daily basis and do statistical analysis on the data to locate trends. You can display results in either tabular or graphical form.

Third, you can tap into an information source to receive up-to-the-minute stock quotations and market information. To use such an information source, you would use the PCjr together with the PCjr Internal Modem, which allows the PCjr to receive information over telephone lines.

Financial Decision-Making

Suppose that you are considering buying a house. The PCjr can help you plan your mortgage commitment. Suppose, for example, that you are considering a mortgage of approximately \$80,000 and that the current interest rate

is approximately 13%. Using the mortgage program on the Sampler diskette that comes with each enhanced PCjr system, you can quickly prepare the following table. It shows the monthly payments for mortgage amounts beginning at \$80,000 and interest rates beginning at 13%. You could take this table with you on a house-hunting trip to determine the effects of varying mortgage amounts and interest rates.

MONTHLY MORTGAGE PAYMENT COMPARISONS										
RATES	30-YEAR MORTGAGE LOAN AMOUNTS					88000	90000	92000	94000	96000
	80000	82000	84000	86000	88000					
13.00	884.96	907.08	929.21	951.33	973.46	995.58	1017.70	1039.83	1061.95	
13.25	900.42	923.13	945.65	968.17	990.68	1013.20	1035.71	1058.23	1080.74	
13.50	916.33	939.24	962.15	985.05	1007.96	1030.87	1053.78	1076.69	1099.60	
13.75	932.09	955.39	978.70	1002.00	1025.30	1048.60	1071.90	1095.21	1118.51	
14.00	947.90	971.59	995.29	1018.99	1042.69	1066.38	1090.08	1113.78	1137.48	
14.25	963.75	987.84	1011.94	1036.03	1060.12	1084.22	1108.31	1132.41	1156.50	
14.50	979.64	1004.14	1028.63	1053.12	1077.61	1102.10	1126.59	1151.08	1175.57	
14.75	995.58	1020.47	1045.36	1070.25	1095.14	1120.03	1144.92	1169.81	1194.70	
15.00	1011.55	1036.84	1062.13	1087.42	1112.71	1138.00	1163.29	1188.58	1213.87	
15.25	1027.57	1053.26	1078.95	1104.63	1130.32	1156.01	1181.70	1207.39	1233.08	
15.50	1043.61	1069.70	1095.79	1121.88	1147.97	1174.06	1200.16	1226.25	1252.34	
15.75	1059.69	1086.19	1112.68	1139.17	1165.66	1192.16	1218.65	1245.14	1271.63	
16.00	1075.81	1102.70	1129.60	1156.49	1183.39	1210.28	1237.18	1264.07	1290.97	
16.25	1091.95	1119.25	1146.55	1173.84	1201.14	1228.44	1255.74	1283.04	1310.34	
16.50	1108.12	1135.82	1163.52	1191.23	1218.93	1246.63	1274.34	1302.04	1329.74	

PRESS SPACE BAR TO CONTINUE

Figure 2-1. A table of monthly mortgage payments prepared with the program MORTGAGE.

Spreadsheets

One of the most popular pieces of software ever created is the VisiCalc¹ program. This program is a financial planning tool that can be used in hundreds of situations, both business and personal. The VisiCalc program was the first example of a class of programs called **spreadsheets**. In this section, we will describe a typical spreadsheet application—a home budget. Suppose that the following data describe the income and spending of the Smith family.

Income	
John	\$1580 per month
Sally	\$1145 per month
Interest	\$ 280 in Jan., Apr., July, Oct.
Bonus	\$3500 in Dec.

¹VisiCalc is a registered trademark of VisiCorp.

Expenses	
Mortgage	\$487.73 per month
Car payment	\$187.50 per month
Utilities	\$207.00 per month
Clothing	\$200.00 per month
House repairs	\$100.00 per month
Food	\$640.00 per month
Entertainment	\$120.00 per month
Fuel	\$135.00 per month
Life insurance	\$583.00 in Oct.
Vacation	\$1,000.00 in Jul.
Health insurance	\$285.00 in Apr. and Oct.
Pension plan	\$400.00 per month

Everyone has probably compiled a list of this sort at one time or another. Once the list is complete, there are a number of important questions to ask, such as:

1. Is the Smiths' income sufficient to cover their expenses?
2. Is there any month in which they will be short of cash?
3. How much is their annual savings (deficit)?

To answer such questions (especially the second one), the best approach is to compile a month-by-month description of income and expenses. Such a description could be written out on an accounting worksheet in the following fashion:

	Jan.	Feb.	Mar.	...	Dec.	Year
Income						
.						
.						
.						
Expenses						
.						
.						
.						
Total						
Income						
Total						
Expenses						
Income						
minus						
Expenses						
Cumulative						

We will leave it to you to fill in the above chart from the given data. If you do, you will discover that the Smiths are not making ends meet. There are a number of months in which they have a deficit, and they are running at a deficit for the entire year. Being sensible people, the Smiths want to recast their budget so that they are not spending more than they are earning, and so that they are not running a deficit in any month.

They begin to experiment with various cutbacks in expenditures. For each proposed cutback, they must rework the above chart to determine if they have erased all the monthly deficits. It might take 15 or 20 tries before they arrive at a budget they can live with that is free of deficits. The calculations involved are quite lengthy if done by hand. However, they are exactly the sort of calculations that a spreadsheet program can handle easily.

A spreadsheet program turns your computer screen into a large accounting worksheet, just like the budget worksheet above. Note that the Smiths' budget required a chart consisting of 14 columns and 20 rows. The typical spreadsheet allows many more rows and columns than that, although the number depends on the particular program and the amount of memory in your machine. Figure 2-2 shows the layout of a typical worksheet display.

For example, in the worksheet of Figure 2-2, the **cursor** (blinking line) position is the first row, first column or position A1.

Our first task is to customize the worksheet to the data given in the Smiths' budget. Let's put the headings in the first row and first column. In our exam-

A1	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										

Figure 2-2. A typical spreadsheet display.

ple, we would move the cursor to position B1 and type “Jan” followed by **ENTER**. We would then move the cursor one position to the right and type “Feb” and so forth. Similarly, we type the row labels beginning in the first column of row 2. Note that the first line of the display always indicates the current position of the cursor.

Let’s now assume that the row and column headings are typed as in Figure 2-3.

	A	B	C
1		Jan.	Feb.
2	Income		
3	John		
4	Sally		
5	Interest		
6	Bonus		
7			
8	Expenses		
9	Mortgage		
10	Car payment		
11	Utilities		
12	Clothing		
13	House repairs		
14	Food		
15	Entertainment		
16	Fuel		
17	Life insurance		
18	Vacation		
19	Health insurance		
20	Pension plan		
21	Total Income		
22	Total Expenses		
23	Income Minus Expenses		
24	Cumulative Total		

Figure 2-3. Worksheet with expanded columns.

We now wish to enter the budget data into the columns Jan.-Dec. Let’s begin with the Jan. column. We must position the cursor at location B3 and type John’s income for January, namely 1580. We then move the cursor down one entry and type Sally’s income for January, and so forth. In similar fashion, we enter all the data in the budget for the month of January.

Let’s now calculate total income and total expenses for January. Total income for January is obtained by adding the sum of the entries B3 through B6. We may let the spreadsheet do the arithmetic for us. (The precise command for indicating this arithmetic operation will depend on the particular spreadsheet program.)

To calculate the total expenses, add up the entries in B9 through B20. By giving an appropriate spreadsheet command, we define the entry in B22 to be the sum of the entries in B9 through B20. Entry B24 is the same as B23. This completes the entry of the column corresponding to January.

To complete the second column, it is not necessary to go through all the trouble we just went through for the month of January. Indeed, all entries are the same, except that the \$\$280 item for interest must be changed to 0. Most spreadsheets have a feature that allows you to duplicate any sequence of entries. We can use this feature to fill in all the entries in the columns C,D,....,M.

The worksheet is now complete except for the final column, N, which records totals, and the final row, which records cumulative totals. Let's first fill in the final column. For example, in position N2, we want the sum of B2, C2, ..., M2.

In the final row, the entry at C24 is $B24 + C23$. That is, the cumulative surplus (deficit) for February is the surplus (deficit) for January (entry B24) plus the surplus (deficit) for February (entry C23). Similarly, the cumulative surplus for each month may be obtained by adding the cumulative surplus for the preceding month to the surplus for the current month. We then insert $B24 + C23$ into position C24 and repeat this formula in positions D24 through M24. This completes the development of the worksheet for the Smith budget.

Once the worksheet has been created, you may alter any entry in it, and all the other entries will be recalculated according to the relationships specified. For example, you can observe the effect of a raise in income, a cut in expenses of a particular type, an inheritance, a tax refund, and so forth.

The idea of a spreadsheet is simple. But it can be an invaluable aid in doing all kinds of financial planning. Once the relationships between the various quantities have been defined, you may vary the values of some quantities and see what happens to the others.

2.4 Word Processing

If you use your PCjr for nothing else but word processing, you will receive ample returns from your investment. That's a pretty strong statement. But once you use your PCjr as a word processor, you'll never use a typewriter again.

Some of you may already know what a word processor is. If you don't, a word processor is a computer program that turns your computer into a fancy typewriter. The fundamental operations of a word processor are to:

- Type a document and store it in RAM.
- Make changes in the document as you type.
- Save a long-term copy of a document on diskette.
- Recall a document from diskette.
- Print a document on the printer.

To type a document, you proceed pretty much as on a typewriter. However, you type a paragraph as a continuous stream of text. Don't worry about where lines end. The word processor will decide. When you reach the end of a line, the word processor will automatically pick up any partial word and move it to the beginning of the next line. You indicate an end-of-line only at the end of a paragraph.

A word processor allows you to change what you have typed. For example, you may add a word, phrase, or sentence in the middle of a paragraph. The word processor will then rearrange the paragraph to accommodate the change. Similarly, you may delete words, phrases, or whole paragraphs.

Most word processors are able to move and copy text. There are several different types of move and copy operations, including:

- Move a block of text to a new location within the current document.

- Copy a block of text to a specified location within the current document (but also leave a copy of the block in its current location).

- Copy a block of text into a diskette file.

- Insert a block of text from a diskette file into a specified position in the document.

In addition, most word processors include search and replace features. For example, you could search the current document for all occurrences of the phrase "less than" and change each to "greater than."

All the above editing takes place within the PCjr's memory. You may inspect your editing changes by displaying any part of the edited document on the screen.

After a document has been edited to your satisfaction, you may print it out on your printer. Your printing instructions may include a number of formatting commands to specify:

- Spacing between lines

- Type style and size

- Top, side, and bottom margins

- Page numbering requirements

- Page headings and footings

- Number of lines per page

Word processors often have a number of auxiliary programs to help you in other aspects of document preparation. Probably the most popular are the spelling checkers that proofread a document against a dictionary and flag suspicious-looking words for you to check. There are also programs to compile tables of contents, indexes, and figure lists.

Some word processors come with document-merging capabilities, which allow you to take parts of a document from a file on diskette. In this way, for example, the word processor could read successive names and addresses from a mailing list on diskette and incorporate them in the appropriate places in a form letter. Imagine how helpful this can be in sending out announcements to members of your church or civic group!

As you might have gathered by now, word processing is practical only in a system with a diskette drive. Also, an 80 character wide screen, although not strictly required, is a big help. And, of course, a printer is a necessity.

IBM is offering the **Home Word** word processing program, which is designed especially for personal use. If you wish, you may use one of the more expensive (and more capable) word processors designed for the PC or PC/XT, such as WordStar², EasyWriter³, or Volkswriter⁴.

2.5 Education

The PCjr can be an ideal teacher: It has infinite patience and teaches without making any judgments. What's more, children love to learn by using the computer!

There are many educational programs for your PCjr, both from IBM and others. Here are some educational areas in which the computer is particularly effective:

Arithmetic—Your children can learn arithmetic facts using timed and graded software exercises.

Typing—There are several good programs to teach typing.

Geography—The graphics capability of the PCjr can be used to draw maps and fill in location names.

Interpreting Charts & Graphs—The graphics capability of the PCjr can be used to draw charts and graphs. Students can learn to translate data into graphical form and vice versa.

2.6 Do-It-Yourself Programming

All of the applications considered so far have used software you buy. However, you can program the PCjr yourself. And what you can accomplish is limited only by your imagination.

The PCjr comes equipped with a cassette version of the BASIC programming language. As an option, you may equip your machine with Cartridge

²Registered trademark of MicroPro International Corp.

³Registered trademark of Information Unlimited Software, Inc.

⁴Registered trademark of Lifetree Software, Inc.

BASIC, which includes all the commands of Cassette BASIC, plus some others.

BASIC on the PCjr is an incredibly rich language that allows you to specify very complex actions with only a few commands.

Learning BASIC is easy and fun. IBM has included a number of tutorial aids with the machine, including:

Keyboard Adventure, a computer-based tutorial explaining the features of the keyboard.

Hands-On BASIC, a beginner's guide to the BASIC language, in book form.

Learning DOS, a beginner's guide to the Disk Operating System, in book form.

A large number of other books and tutorials on PCjr programming will be offered by various other companies.

2.7 Communications

You may equip the PCjr with an internal modem that lets you communicate with other computers. Using this communications capability, you may:

Exchange files with a computer at your place of business.

Let the PCjr act as a terminal of another computer.

Tap into one of the many available subscription information services, such as the Dow Jones News Service⁵ or The Source⁶.

Communicate with other computer enthusiasts via one of the many available bulletin board services. (In case you are not tuned in to this trend, a **bulletin board** is a computer set up to receive and pass on messages, usually without operator intervention.)

The PCjr internal modem is an inexpensive way to connect your computer to the outside world.

2.8 Graphics and Sound on the PCjr

The PCjr is capable of quite sophisticated color graphics. You can shed light on pages of financial statistics by expressing the same information in the form of a bar chart or pie chart, as in Figures 2-4 and 2-5.

⁵Registered trademark of Dow Jones, Inc.

⁶Registered trademark of Source Telecomputing Corporation.

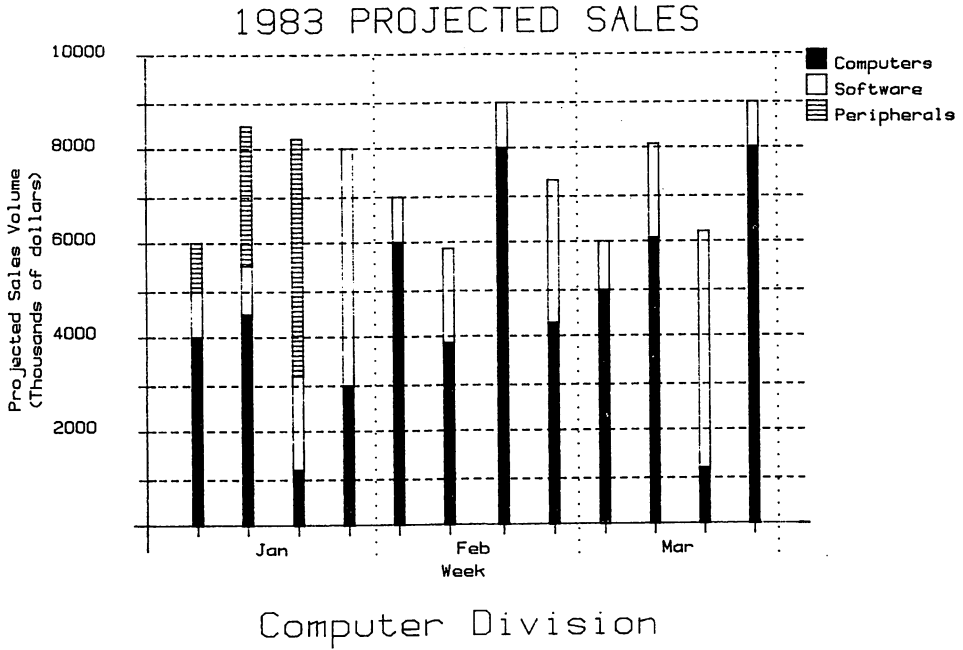


Figure 2-4. A bar chart drawn by using PC BASIC.

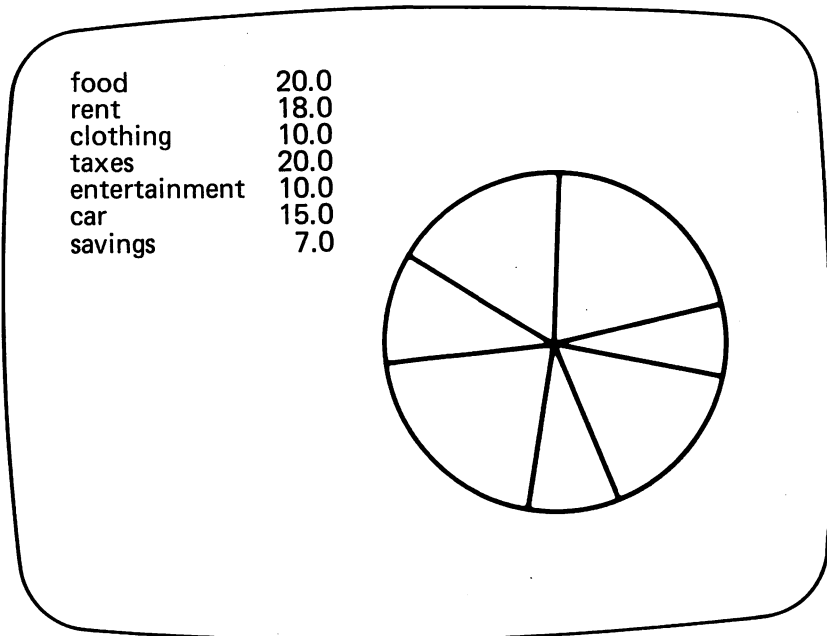


Figure 2-5. A pie chart drawn by using PC BASIC.

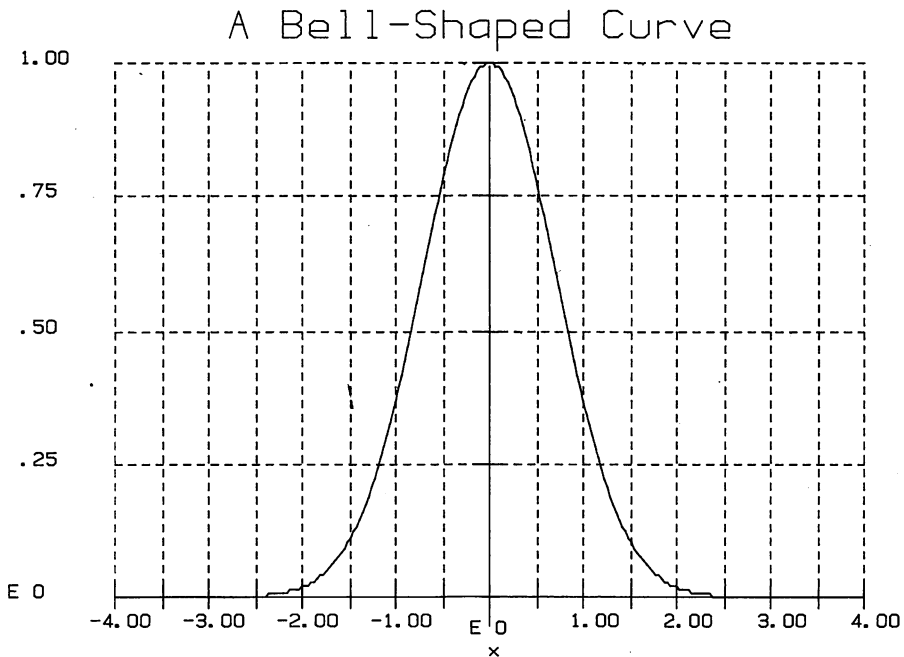


Figure 2-6. A bell-shaped curve.

You can even prepare graphs of the functions you meet in a high school algebra course. For example, Figure 2-6 is a graph of a bell-shaped curve drawn with PC BASIC.

Not all graphics must have a serious purpose, however. You can use the graphics capabilities of the PCjr to draw computer art, as in Figures 2-7 and 2-8.

You can even print out copies of graphics displays on a printer or plotter. The printed graphs can be used in your business, for a club or committee, or a school project.

PCjr BASIC has commands that let you produce animations. My son has produced some great animation, including rockets exploding and birds hatching from eggs. In producing them he has learned an incredible amount about programming. (He says he would rather program games than play with them on his video game system!)

The PCjr has a sophisticated sound system based on a music synthesizer chip built into the system unit. Using this chip and a special musical language⁷, you can program the PCjr to play songs in as many as three voices. Experiment with harmonies, compose music, or just transcribe your favorites to use as background music for your programs. You are limited only by your own musical creativity.

⁷Registered trademark of Microsoft, Inc.

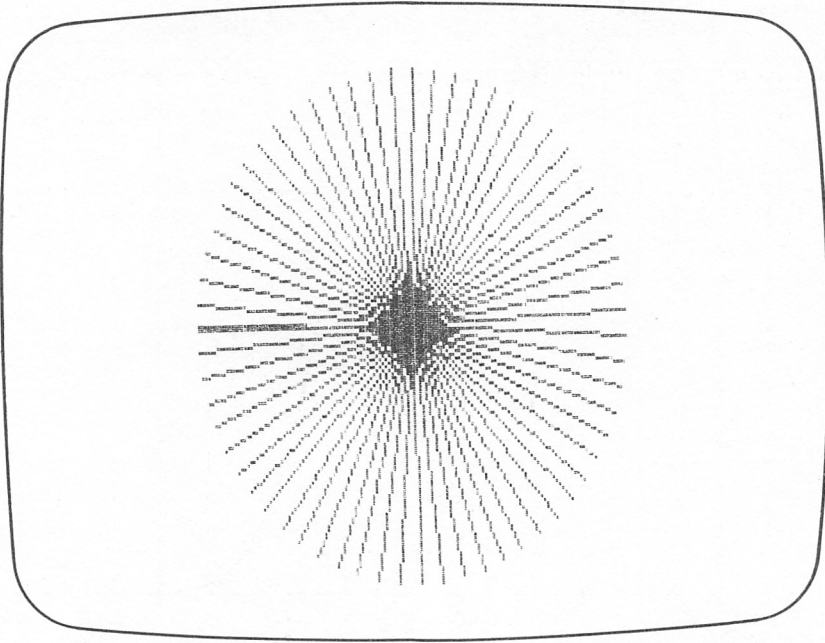


Figure 2-7. Computer art.

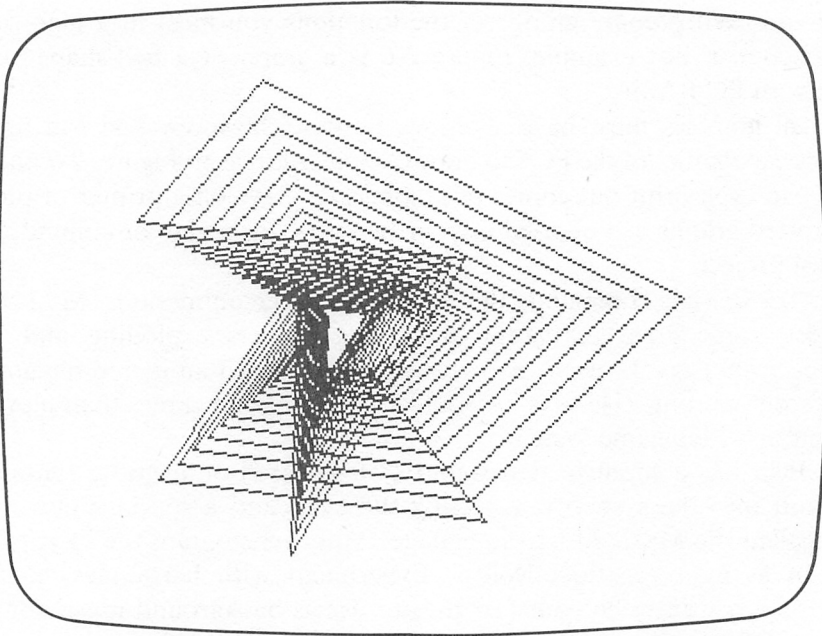


Figure 2-8. More computer art.

2.9 Conclusion

Did we mention any applications that you are interested in? I hope so. There are so many applications that we couldn't possibly even begin to describe them all.

In what follows, you should keep your particular interests in mind as we discuss the hardware. Because you will have to choose your system components (no prepackaging here!). And your interests will be of assistance in structuring your system.

3

A CLOSE LOOK AT THE PCjr's HARDWARE

Now that we have taken a quick tour of the PCjr Home Computer, let's take a detailed look at each of the system components.

3.1 The Keyboard

The keyboard is the device you use to communicate directly with the computer. As you type, your keystrokes are recorded in RAM, awaiting action by the computer. What you type is called **input**.

The PCjr keyboard is very similar to that of a typewriter. However, it contains a number of special keys that are not found on a typewriter keyboard, allowing it to perform many more functions. The keyboard of the PCjr is shown in Figure 3-1.

At first, the PCjr keyboard may seem strange and forbidding. However, you can easily learn about its various keys using the *Keyboard Adventure* tutorial that is stored in ROM. By using a certain procedure when you turn the PCjr on, you start *Keyboard Adventure*. This is one of several tutorials that come with the PCjr.

The keyboard contains electronics to transfer your keystrokes to the system unit. This may be done in either of two ways:

Infrared Optical Link—The keyboard has a built-in infrared communications system that can transmit information from the keyboard to the system unit, without using a cord. This system is powered by four AA size batteries and lets you type on the keyboard as much as 20 feet from the system unit. It is possible for you to use the keyboard while sitting in an easy chair across the room from the system unit and display screen. The cordless keyboard also gives you great flexibility in arranging your system in crowded rooms. For instance, the system unit could be placed on a shelf in a wall unit and the keyboard across the room near the television.

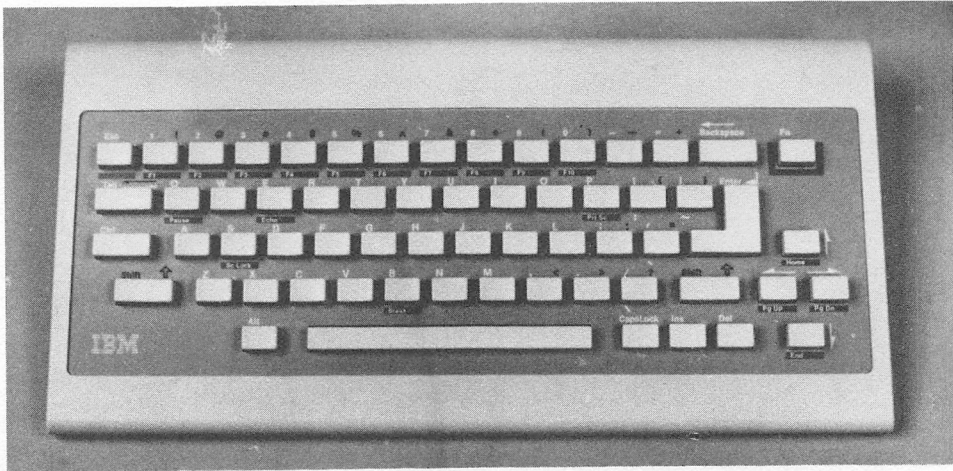


Figure 3-1. The PCjr keyboard.

The infrared optical link has one restriction. It cannot “turn corners.” There must be an unobstructed straight-line path from the system unit to the keyboard.

Cable Connection—You may connect the keyboard to the system unit via an optional cable. This is advisable in classroom situations, or any case where several PCjr's are in one room.

3.2 System Unit

Of all the components of the PCjr system, the system unit is the most complex. Within its small cabinet is all the ingenious electronic circuitry that makes the computer work. In this section, let's get familiar with the system unit, first from the outside, and then on the inside.

Let's begin by looking at the front of the system unit in Figure 3-2.

At the bottom right are two slots into which you may insert program cartridges. Initially, IBM is offering assorted program cartridges containing games. In addition, you may purchase Cartridge BASIC, an enhanced version of the Cassette BASIC permanently stored in the PCjr's memory.

Actually, the cartridges are Read Only Memories (ROM's). By plugging in a cartridge, you are extending the PCjr's memory by 64k. However, you can't write in cartridge memory. You can only read it.

The system unit may contain a diskette drive. (We'll discuss how to use a diskette drive later on.) If the diskette drive is installed, you will find it at the top right of the system unit, as shown in Figure 3-2. The enhanced model of

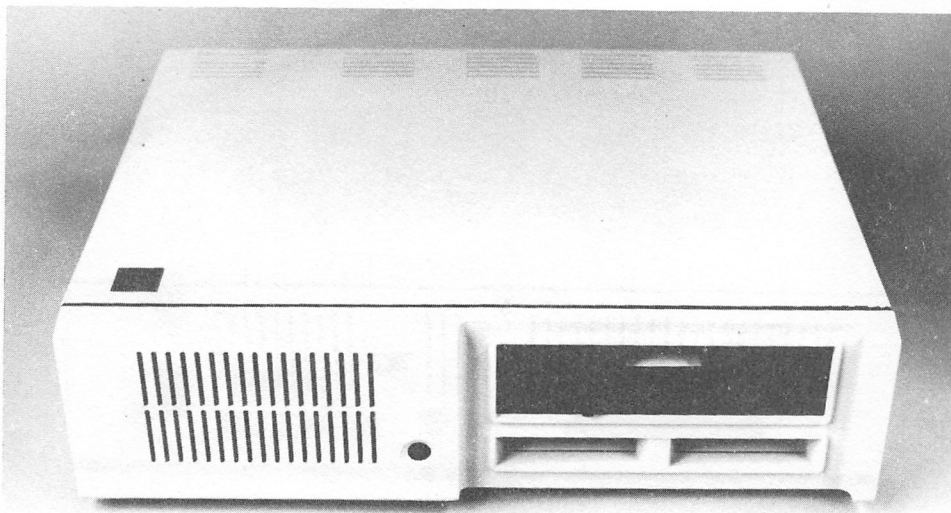


Figure 3-2. The front of the system unit.

the PCjr comes with a diskette drive as standard equipment. However, even if you buy the basic model, it is very simple to add a diskette drive later.

If you look at the sides and rear of the system unit, you will notice a number of plugs (“connectors” in computerese). (See Figs. 3-3 and 3-4.) These allow you to connect a number of different devices to the system unit.

The connectors are coded with letters. For instance, the connector into which you can plug a display device is labeled “D.” The various connectors on the PCjr include:

Display—You must connect the display to the system unit via a cable. The type of cable and the connector into which it is plugged will depend on the display you choose. (See Section 3.4.)

Transformer—All electrical power for the system unit passes through the transformer. So the transformer must be connected to the system unit via a cable.

All systems require the display and transformer cable connections. In addition, your system may require some of the following connections, depending on your system configuration.

Keyboard—As we mentioned in Section 3.1, you may connect the keyboard to the system unit via a cable. If you choose this option, then the cable plugs into a connector at the rear of the system unit.

Printer—If you have a printer, you connect it to the system unit via a cable. Just which plug you use will depend on the printer you are using. See Section 3.5.

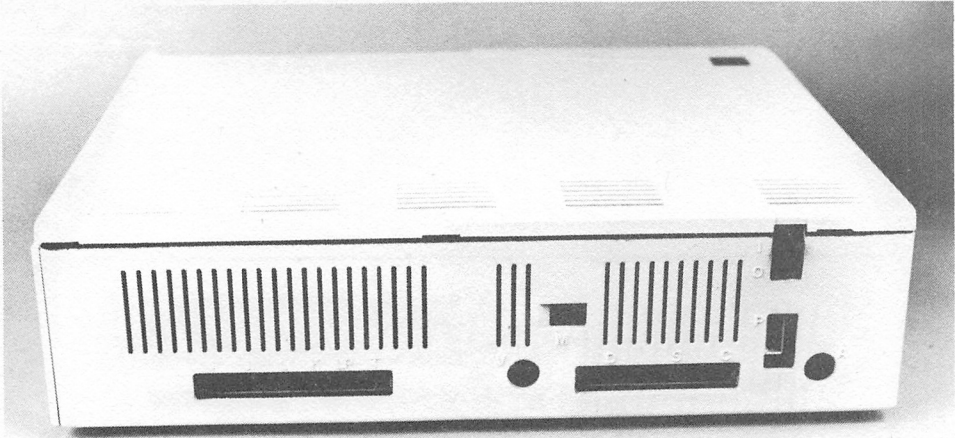


Figure 3-3. The rear of the system unit.

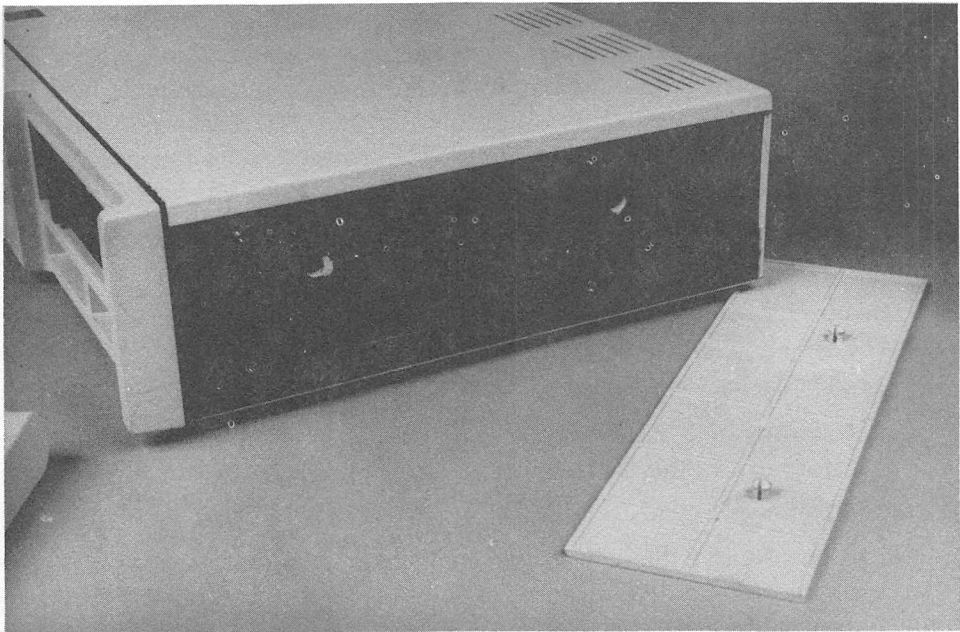


Figure 3-4. The sides of the system unit.

Joystick—You may equip your computer system with an optional joystick. (See Fig. 3-5.) The joysticks you use with a personal computer are similar to those used with the home video games which have become so incredibly popular. The system unit already contains the circuitry necessary to use the joystick, which you plug into the rear of the unit.



Figure 3-5. A joystick.

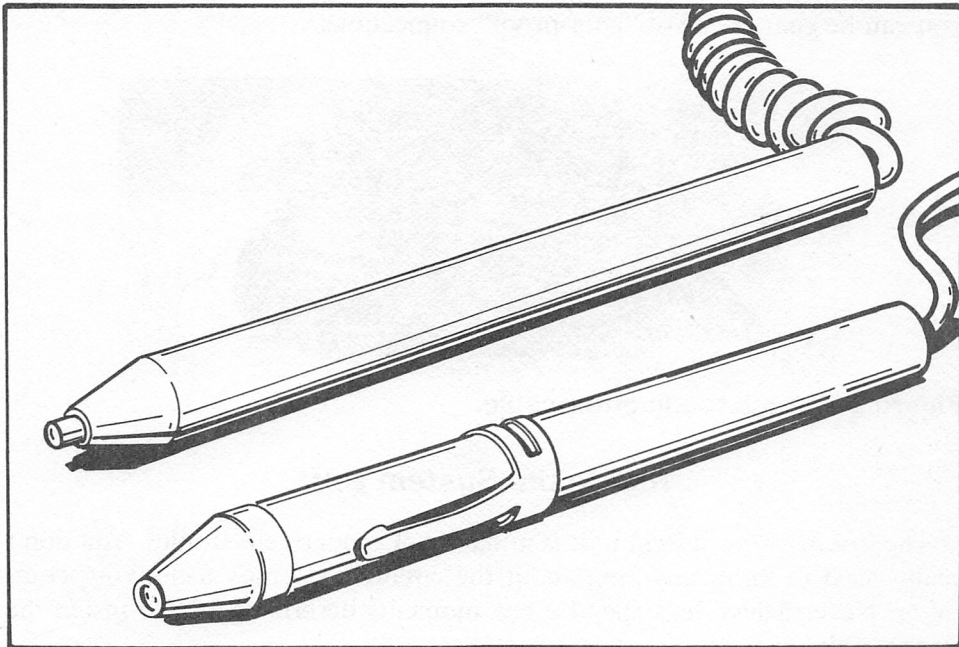


Figure 3-6. Light pens.

Light Pen—A light pen is a device that lets you input information just by pointing to the screen. (See Fig. 3-6.) For example, the computer might display a list of actions. You could use a light pen to point to the action you wish the computer to take. The light pen circuitry is able to tell where the tip of the light pen touches the screen. You may plug a light pen into the rear of the system unit.

Modem—A modem is a device that translates internal computer data into electric signals that can be transmitted over a phone line. Using a modem, your PCjr can communicate with another computer or a data service (more about these later). If you add a modem to your PCjr, you'll need to connect the cable that comes with the modem to a modular phone jack.

More About Cables—You may be confused by the above discussion of cabling requirements. Don't be. The PCjr cables are designed so that they can only be connected to the correct plug. For example, look at the plug in Figure 3-7. It contains 25 pins. Each pin carries an electrical signal with a particular meaning. So it's important that each pin be plugged into the correct place. Note that the end of the plug is not rectangular. The mating connector has the same shape. The unusual shapes mean that you can make the connection in only one direction. Therefore, each pin will be plugged into the correct place. Each of the PCjr's cables has a different shaped plug so that you can be guaranteed of "goof-proof" connections.

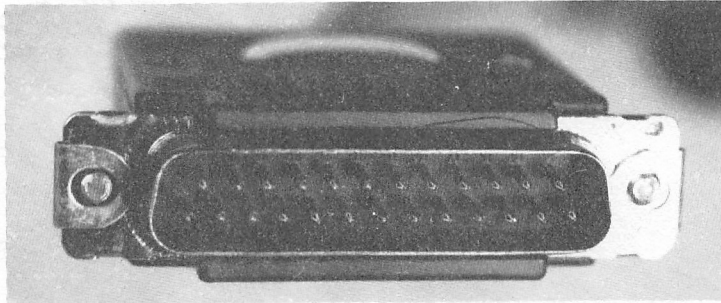


Figure 3-7. A microcomputer cable.

Inside the System Unit

The inside of the system unit is a marvel of modern electronics. You don't really need to know anything about the circuitry to enjoy using your computer. Nevertheless, let's spend a few moments describing what's inside the system unit.

Circuit Boards—Most of the electronic circuitry of the computer is organized on circuit cards like the one shown in Figure 3-8. A circuit card is usually called a "board."

A circuit board contains pathways along which electricity can flow. These pathways connect various semiconductor devices (also called "chips") to one another. There are various types of chips on the circuit boards of your PCjr, but the most important is the 8088 microprocessor, which is the "brain" of the computer. The 8088 is approximately an inch long, but contains tens of thousands of transistors that control the inner workings of the computer.

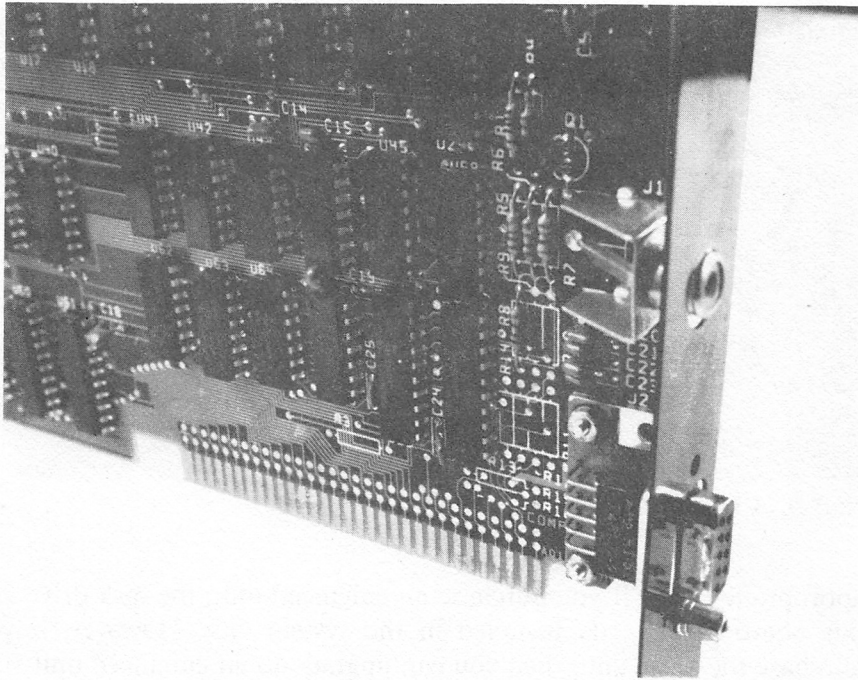


Figure 3-8. A circuit board.

RAM Memory—On the circuit boards of the system unit are memory chips, both ROM and RAM. To get an idea of the size of such a chip, look at Figure 3-9, which shows a RAM chip beside a paper clip. RAM memory is used for many purposes. When you type in a BASIC program, it is stored in RAM. There are sections of RAM devoted to communications with the printer, screen, and keyboard.

ROM Memory—Some of the memory chips inside the computer are ROM's, or Read Only Memories. These chips contain information that has been pre-recorded at the factory and which you are unable to change by normal operation of the computer. Within the ROM's are various diagnostic routines, a tutorial to help you learn the keyboard, and the computer language Cassette BASIC. The cartridges that you plug into the slots on the front of the system unit are just ROM's that extend the ROM memory otherwise available within the system unit.

Expansion Slots—Inside the system unit, on the main circuitry board, are three slots into which you may insert additional circuit boards. The slots are designed so that each will take only one type of board. Here are the possibilities:

Diskette Drive Adapter Board—This board contains the electronics necessary to control a disk drive. To use a disk drive, this card must be in the appropriate expansion slot and connected to the disk drive via an

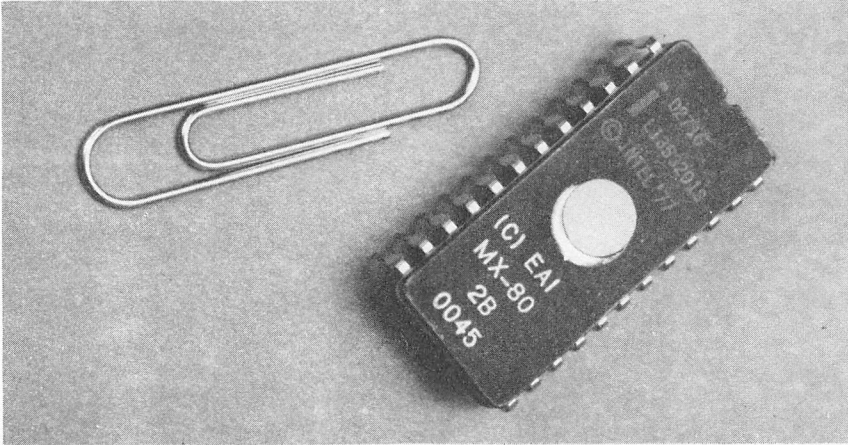


Figure 3-9. A RAM chip.

appropriate cable. If you purchase an enhanced unit, the disk drive and this board are already installed in the system unit. However, if you purchase the basic unit, then you can upgrade to an enhanced unit simply by adding the extra circuit board and disk drive.

Internal Modem—This circuit board allows you to connect your computer to a telephone jack to establish computer-to-computer communications over standard telephone lines.

64k Memory Expansion and Display Adapter Board—This board increases total RAM to 128k and allows you enhanced display capabilities, including:

- * Choice of 80 character line width.
- * Use of high-resolution graphics.

(For a further discussion of display capabilities, see Section 3.4.) This expansion board comes standard in the enhanced system unit. It may be added to the basic system unit.

Other Possibilities—The IBM PC has five expansion slots; the PC/XT has eight. By comparison, the PCjr is very short on expansion slots. Before you commit yourself to filling all your slots, you should consult your dealer on the latest offerings from other companies. If the past is any guide, there will be available a large number of multi-purpose expansion boards that incorporate 3, 4, or more functions on a single board. By using such boards, you can make most efficient use of your precious expansion slots.

3.3 Cassette and Diskette Storage

The PCjr can use either cassettes or diskettes for long-term storage. You will require one or the other of these devices with your PCjr system if you are to be spared the chore of repeatedly typing the same programs and data each time you wish to use them. (Recall that RAM is erased when the computer is turned off.)

Cassette Recorder

You may connect a standard cassette recorder (see Figure 3-10), via an optional cable, to one of the connectors at the rear of the system unit. Cassette BASIC, the version of the BASIC language that is in ROM and automatically comes with every PCjr, contains instructions to save and recall programs from cassette. In addition, Cassette BASIC allows you to read and write data files (like address lists, form letters, financial data) to and from cassette.

As your storage medium, you may use the same cassettes you now use to tape music or conversations. Just stay away from the bargain-basement cassettes. A small flaw in a tape will show up as static in a song. However, it

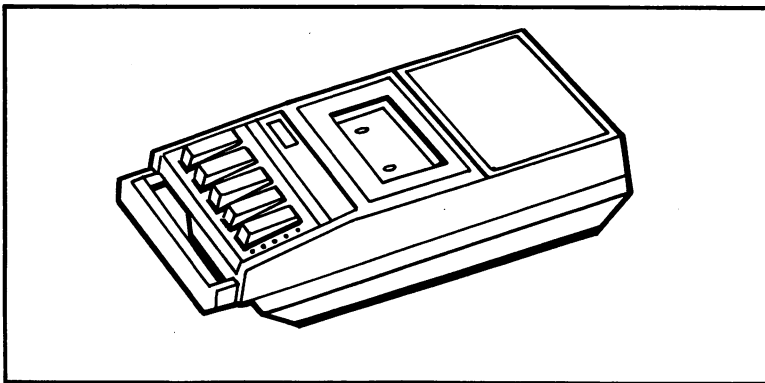


Figure 3-10. A standard cassette recorder.

could spell disaster for a stored computer program or a crucial data file. If you stick to good quality tapes, however, you should have no problems.

Note that your PCjr comes with all the electronics necessary to read and write cassettes. The only additional components you must add are the cassette recorder itself and the optional cable purchased from IBM (and, of course, a supply of cassettes).

Diskette Drive

You may use a diskette drive with the PCjr. If you have purchased a basic unit, then you may add the diskette drive later. However, if you purchased the enhanced unit, then a diskette drive is standard.

The visible portion of the diskette drive is the door at the bottom of the front of the system unit. This door is for inserting diskettes (more about that later). The electronic circuitry necessary to operate the diskette drive is contained on the **diskette drive adapter** that occupies one of the expansion slots inside the system unit.

The diskette drive for the PCjr is a “half-height” drive and can write either single-sided or double-sided diskettes. It uses “double-density” diskettes and can write 160k bytes on each side of the diskette. We’ll discuss the detailed operation of the diskette drives in a later chapter.

3.4 Displays

The display of your PCjr system is an extremely important component. As you type on the keyboard, the keystrokes are displayed on the screen. (In computer jargon, the keystrokes are **echoed** to the screen.) In addition, the display is an output device to which programs send data.

You have three choices for the display of your system:

A television set

A monitor

The IBM color display

Let’s describe the features of each of these displays:

Television Set—You can use a television set (black and white or color) as a display. To connect it to the PCjr, you will need the optional PCjr Connector for TV. You have probably used a similar connector if you have ever connected a video game system to your television. The connector attaches to the video input of your television (usually via two screws on the rear of the set). It has two positions: TV and Computer.

A television set allows you to display 25 lines of 40 characters each. In graphics mode, a television screen is divided into 200 lines of 320 dots each.

In the case of a color set, you may display color text and graphics. Moreover, the PCjr can produce sound using the speaker system in the television set.

Monitor—A monitor resembles a television set, except that it cannot receive regular commercial broadcasts. It can display only what is sent to it from the system unit. Monitors may be either color or black and white.

The PCjr can use any monitor that accepts what is called a “standard composite video signal.” Don’t worry what this means. If in doubt, ask your dealer if a particular monitor will work. The display on a monitor will generally be clearer than a corresponding display on a television set. Moreover, if your system is equipped with a **Memory Expansion and Display Adapter** board, then a monitor will give you additional display capabilities as follows:

Text lines may be either 40 or 80 characters wide.

You may use the **high-resolution graphics mode**, in which the screen is divided into 200 lines of 640 dots each. However, you should note that in high-resolution graphics mode, you may use only four colors simultaneously, as compared to eight in medium resolution and 16 in low-resolution mode.

A monitor will require a special cable to plug it into the connector on the display adapter. (Note that the adapter cable for a television set will not do the job.)

IBM Color Display—This display is a color monitor that produces particularly striking colors. You may use this display on any of IBM’s personal computers (PCjr, PC, or PC/XT). To use it on the PCjr, you will need a special cable to connect it to the system unit.

3.5 Printers

There are several printers available for use with the PCjr. They span quite a wide range of printer technology as well as a wide price range. Let’s discuss them in order of increasing price.

IBM Compact Printer

This is an inexpensive printer that takes up only a small amount of space. It forms characters from dots, with each character printed in a 5-dot x 8-dot rectangle.

This printer can print 50 characters per second and is capable of printing a total of 191 different characters (the usual letters, numbers, and punctuation marks, as well as a set of graphics characters).

The IBM Compact Printer uses thermal paper, which you may purchase in individual sheets, rolls, or fanfold.

You may print an 8-inch line on standard 8-1/2 inch wide paper. The printer can print standard width characters (10 per inch), double-width characters (5 per inch), compressed characters (17.5 per inch) and compressed, double-width characters (8.75 per inch).

The IBM Compact Printer has a graphics mode in which you may control the individual dots in each character rectangle. In this mode, you can print lines which are 560 dots wide and 8 dots high. This corresponds to a resolution of 70 dots per inch horizontally and 48 dots per inch vertically.

The IBM Compact Printer plugs directly into the system unit via a cable that comes with the printer.

IBM 80 CPS Dot Matrix Printer

This is the next printer in order of sophistication and price. It still prints characters as rectangles of dots. However, each character is formed in a 9 x 12 rectangle. The advantages of this printer are:

Its speed is 80 characters per second.

It delivers better print quality than the IBM Compact Printer.

It can use ordinary paper as opposed to thermal paper.

The main disadvantages are:

It takes up more space.

It has no graphics capability.

IBM Graphics Printer

The IBM Graphics Printer is similar to the IBM 80 CPS Dot Matrix Printer, except that graphics capabilities are included. This printer has several graphics modes, allowing much higher resolution than possible on the Compact Printer.

The IBM 80 CPS Dot Matrix Printer and the IBM Graphics Printer both require a **parallel interface**. To supply this interface in the PCjr, you must attach the Parallel Printer Attachment to the right side of the system unit. You then plug your printer into the connector on the attachment.

A detailed comparison of these three printers is given in Chapter 5.

3.6 Communications

Very significant among the features of the PCjr is its communications capability. You may equip your PCjr system with a device called a **modem** that will enable you to transfer computer data over ordinary phone lines. You can

use this capability in many interesting applications, which we'll explore in the next chapter.

Internal vs. External Modem

With the PCjr, you may use either an internal modem, which fits into one of the expansion slots, or you may use an external modem that connects to the serial port (RS232-C interface) at the rear of the system unit.

The PCjr Internal Modem—The PCjr has an expansion slot designed to take the **PCjr Internal Modem** board. This board contains the complete electronics for converting the electrical signals that represent data within the computer into corresponding electrical signals, which can travel along ordinary phone lines and then be reconverted into computer data at the receiving end of the line. (Of course, the conversion at the receiving end must be done by another modem.)

The PCjr Internal Modem has a single cable that extends from the system unit. This cable must be plugged into a modular phone jack. Most phone installations done in the past 10 years use modular jacks. If you can unplug your phone from the outlet in the wall, then the socket remaining after you unplug the phone is a modular jack. To connect the modem, just insert the plug at the end of the cable into the modular jack. You will hear a slight “click” as the plug seats itself into the jack.

PCjr BASIC has the instructions for operating the modem. These instructions allow you to dial a number or repeatedly dial a number until a connection is established. (This is useful for calling a line that is currently busy.)

The speed of data transmission is measured in **baud**. Roughly speaking, 10 baud corresponds to one character per second. So, for example, a transmission speed of 110 baud corresponds to about 10 characters per second. The PCjr Internal Modem gives you a choice of two transmission speeds, 110 and 300 baud.

Using an External Modem—You may also connect an external modem to your PC. To do this, you will need a modem cable to run from the serial port at the rear of the system unit to the modem. There are two types of external modems. The older (and less expensive) type is the **acoustic modem**. This type of modem uses the acoustics in the phone receiver as part of the transmission process. To use an acoustic modem, you must rest the phone receiver in a special cradle on the modem. The computer data is then converted into audible tones that are then converted, by the receiver, into electrical signals to travel along the phone lines. An acoustical modem can transmit at speeds up to 300 baud. An acoustical modem is a possibility if you don't have modular phone jacks.

The more modern kind is a **direct-connect** modem, which connects directly to a modular phone jack, just like the PCjr Internal Modem. A direct-connect

modem can transmit at speeds up to 1200 baud. If you will be transmitting many lengthy documents, you will find that 300 baud is agonizingly slow. In this case, you should consider the added speed of a direct-connect modem. (At 300 baud, it takes about 10 seconds to transmit a single 300-word page. Imagine transmitting a 300-page report!)

Setting Communications Parameters

In order for two computers to communicate, their transmissions must have the same characteristics. In particular, they must be set for the same baud rate. BASIC has the software for adjusting the baud rate, as well as a number of other communications parameters.

3.7 The Serial Port

There are many devices you can connect to your PCjr. And other companies are sure to increase the number of such devices in the months ahead. Most external devices are connected to the PCjr via the serial port, whose connector can be found on the rear of the system unit.

A serial port (also called an RS232-C interface) uses a 25-pin connector, where the electrical signal carried by each pin of the connector is defined by an international standard.

Using the serial port you may, for example, connect some letter quality printers and plotters to your PCjr. Moreover, you may use the serial port to directly wire two PCjr's together so that they may exchange information.

In spite of the international standard of the RS232-C interface, there are many variations requiring special cables. So in using the serial port, you will probably require a special cable for each device to be connected.

BASIC contains instructions for communicating data to the serial port.

4

SOFTWARE & PUBLICATIONS FOR THE PCjr

In this chapter, we discuss the current IBM software offerings for the PCjr. We begin the chapter with a discussion of PCjr BASIC and DOS 2.1. In Section 4.6, we give a summary of the various applications programs currently available from IBM for the PCjr. Section 4.7 summarizes the IBM publications available about the PCjr.

4.1 BASIC on the PCjr

The PCjr comes with two versions of the BASIC programming language: Cassette BASIC and Cartridge BASIC.

Cassette BASIC is stored in ROM within the system unit and is always present on the machine. Cartridge BASIC is an enhanced language and resides on a cartridge that plugs into one of the two cartridge slots at the top of the system unit.

Both versions of BASIC are **interpreted** languages. That is, they run programs by interpreting them one line at a time.

Cassette BASIC provides the instructions necessary to save and recall programs and data from cassette. However, Cassette BASIC provides no disk-related instructions. Here is how to activate Cassette BASIC: Make sure that no disk is in the disk drive and that Cartridge BASIC is not inserted into a cartridge slot. Turn on the system power. The computer will automatically activate Cassette BASIC. Once Cassette BASIC is loaded, you can switch to Cartridge BASIC simply by inserting the cartridge. However, this *resets* the machine. If you ultimately wish to save any programs or data on diskette, be sure to start the machine with Cartridge BASIC in a slot and DOS in the diskette drive.

Cartridge BASIC provides, in addition to all the instructions of Cassette BASIC, the following:

Instructions to control input and output from the diskette (when used with DOS 2.1).

All instructions in Advanced BASIC 2.00 for the IBM PC and PC/XT.

A terminal emulator mode that simplifies communications handling from BASIC.

Ability to load and run a program on cartridge from the second cartridge slot.

A Graphics Macro Language¹ that simplifies programming of complex graphics displays, including animation.

A new Music Macro Language that supports programming the three-voice sound chip.

You may use Cartridge BASIC in conjunction with DOS (see Section 4.5) or on its own. The latter may be accomplished by starting the computer with Cartridge BASIC in either cartridge slot and no diskette in the disk drive. Of course, if you start Cartridge BASIC without DOS, then you won't be able to use the various disk-related BASIC instructions. So if you plan to use a diskette to store your program or data, start your system using DOS.

Cartridge BASIC on the PCjr is one of the richest versions of the BASIC language ever produced. It contains many features not found in any other BASIC. Using Cartridge BASIC, you may incorporate rather complicated effects expressed with amazing simplicity through high-level commands. To give you an idea of the incredible power of PC BASIC, let's consider some examples of BASIC statements that accomplish unusual things.

4.2 Unique Features of PC BASIC

If you plan to do your own programming, it's a good idea to survey the BASIC language of the computer you are getting. Although all home computers come with some version of BASIC, some versions are more powerful than others, in the sense that they have more of the high-level statements that tend to make programming easier.

PC BASIC is one of the most powerful BASICs that currently exist. It would take an entire book to adequately explain even a majority of the statements supported in PC BASIC. So rather than try any comprehensive discussion of that sort, let's just discuss some of the most powerful and unique features.

¹ Graphics Macro Language is a trademark of Microsoft, Inc.

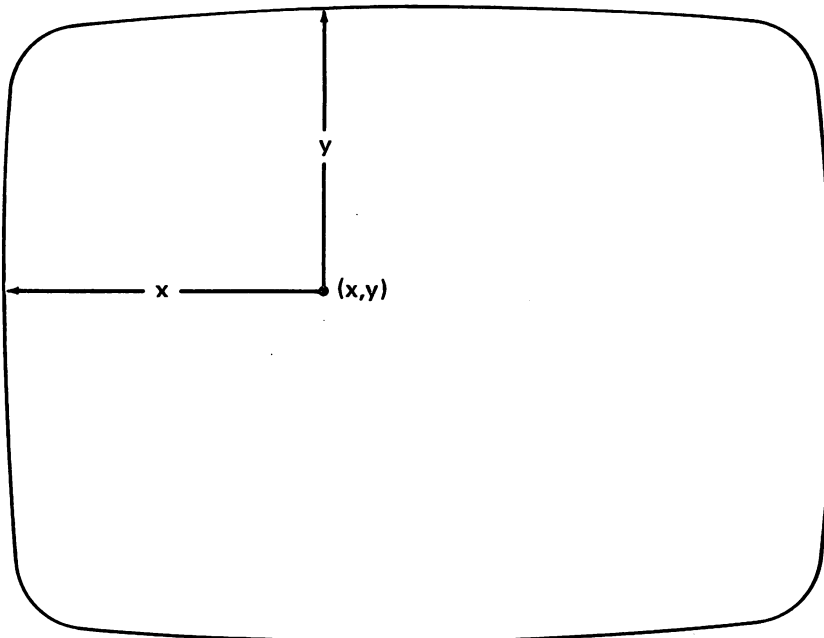


Figure 4-1. Graphics coordinates in PC BASIC.

Colors and Graphics Modes

PCjr BASIC has six graphics modes. In **low-resolution graphics mode**, the screen is divided into 200 rows of 160 dots each. Up to 16 colors can be used. In **medium-resolution graphics mode**, the screen is divided into 200 rows of 320 dots each. As many as 8 colors may be used. In **high-resolution graphics mode**, the screen is divided into 200 rows of 640 dots each. As many as 4 colors may be used. For each resolution, you may enable colors or limit displays to black and white. Each point on the screen is identified by a pair of numbers (x,y) , where “ x ” is the column number and “ y ” is the row number. (See Fig. 4-1.) Column and row numbers start at 0. So in medium-resolution mode, the rows are numbered from 0 to 199 and the columns from 0 to 319. In high-resolution mode, the rows are numbered from 0 to 199 and the columns from 0 to 639.

The background color can be chosen from a set of 16 colors by using the **COLOR** statement. Foreground colors may be chosen from a palette of four colors, one of which is the background color. The colors of the palette may be chosen from among the 16 colors with the **PALETTE USING** statement.

Each point on the screen may be individually colored in any color of the current palette using the **PSET** statement. However, for most applications, it is more convenient to draw using one of the high-level graphics statements discussed below.

Drawing Geometric Figures

The LINE statement allows you to draw straight lines and rectangles. For example, consider the statement:

```
LINE (20,50)-(80,199),1
```

It draws a line connecting the points (20,50) and (80,199) in color 1 of the current palette (see Fig. 4-2).

By specifying a further parameter in the LINE statement, you may draw lines of varying styles, some of which are illustrated in Figure 4-3.

By using the letters B (= Box) and BF (= Box Filled) with the LINE statement, you may draw rectangles and filled-in rectangles. For example, consider the statement:

```
LINE (50,100)-(90,175),1,B
```

It draws a rectangle whose opposite corners are at (50,100)-(90,175). (See Figure 4-4.) If B is replaced by BF in the last statement, the rectangle will be filled in. (See Figure 4-5.)

The CIRCLE statement may be used to draw circles, ellipses, circular and elliptical sectors, both with and without sides. Examples of the output of the CIRCLE statement are shown in Figures 4-6 to 4-8.

The PAINT statement allows you to color regions of the screen. Just specify a point within the region, the color the region is to be painted, and the color of the boundary. PAINT will start at the designated point and paint the region the given color. In this way, you may color interiors of circles, triangles, or any other regions you have drawn.

In painting a region, you may use various paint styles, as illustrated in Figure 4-9.

Windows and World Coordinates

The BASIC VIEW instruction allows you to define a viewport in the screen. All subsequent graphics commands will plot only points within the viewport.

You may use the WINDOW statement to redefine the screen coordinate system to your liking. For example, consider the horizontal direction. The usual screen coordinates run from 0 (= left) to 320 (=right) in medium-resolution mode. You can redefine these coordinates to run from -1 to 1 or -5 to 50. Similarly, you may redefine the vertical coordinates. The graphics commands always refer to points in current coordinates.

You may even combine the effects of VIEW and WINDOW to specify your own set of coordinates within a viewport. This combination can be a real timesaver in drawing graphs, since all of the required coordinate conversion from real-world numbers to screen positions can be done automatically.

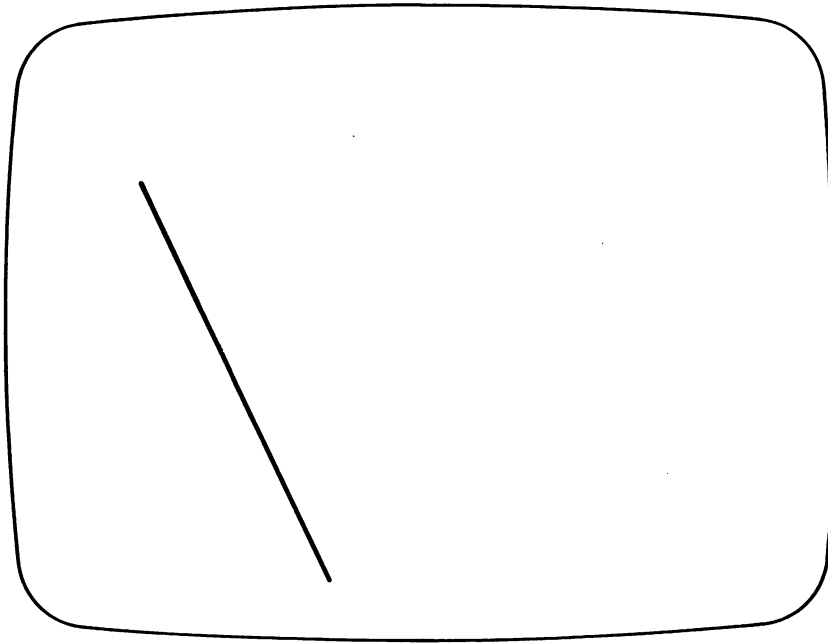


Figure 4-2. LINE (20,50)-(80,199),1

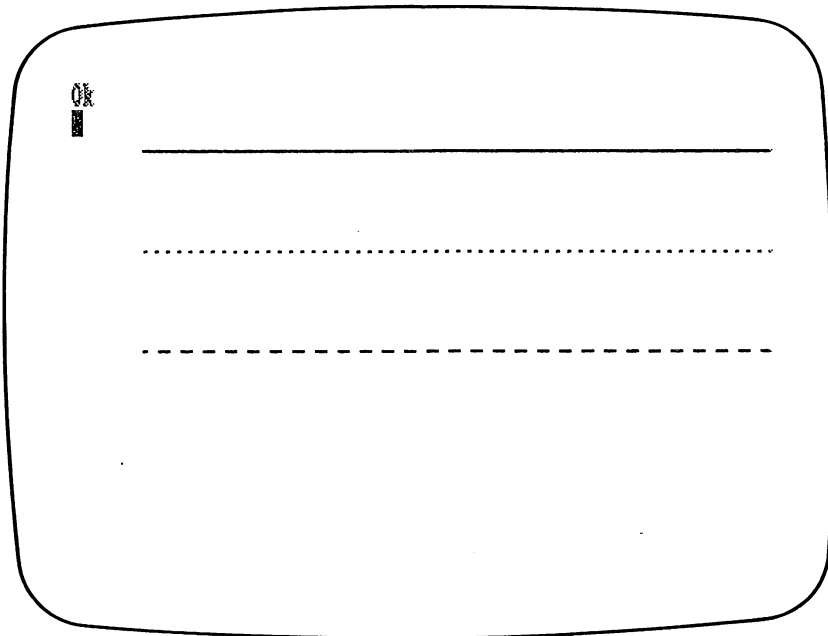


Figure 4-3. Lines of various styles.

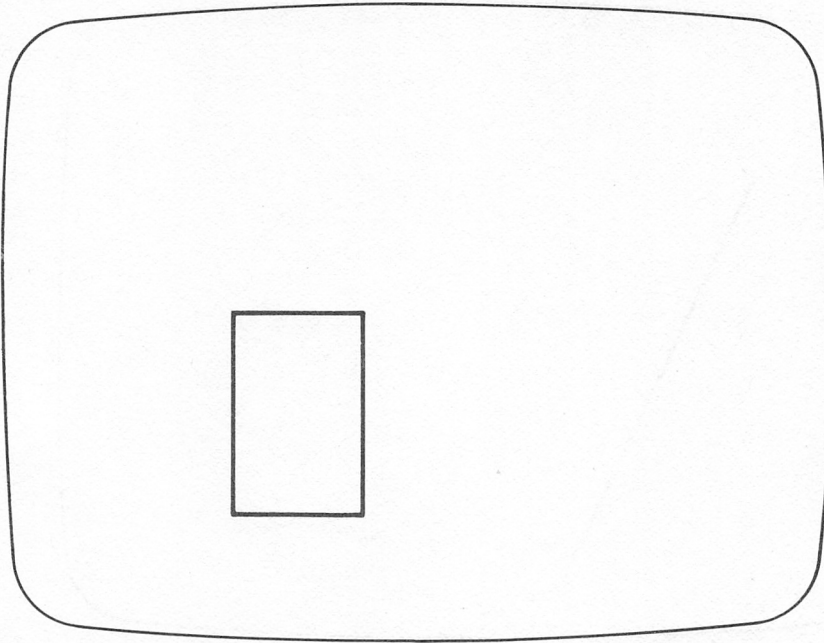


Figure 4-4. The B option.

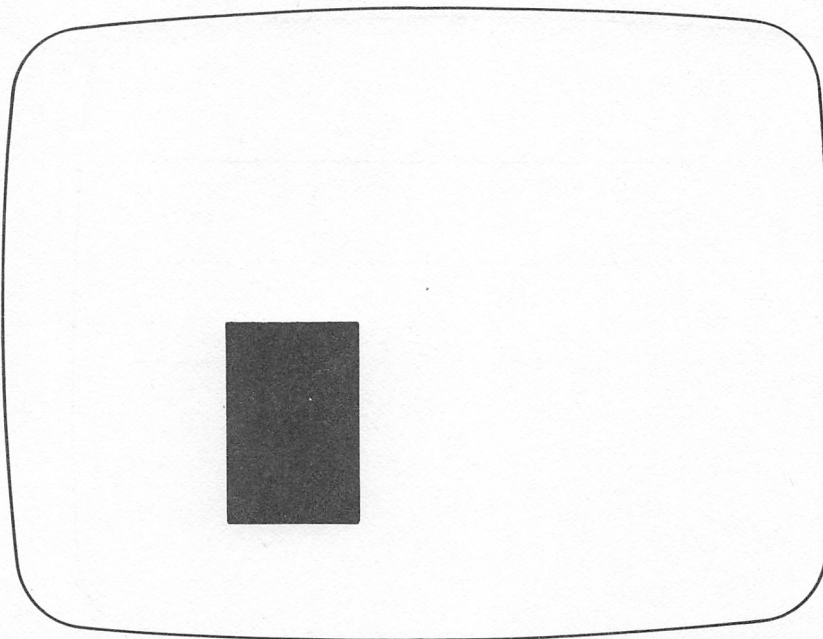


Figure 4-5. The BF option.

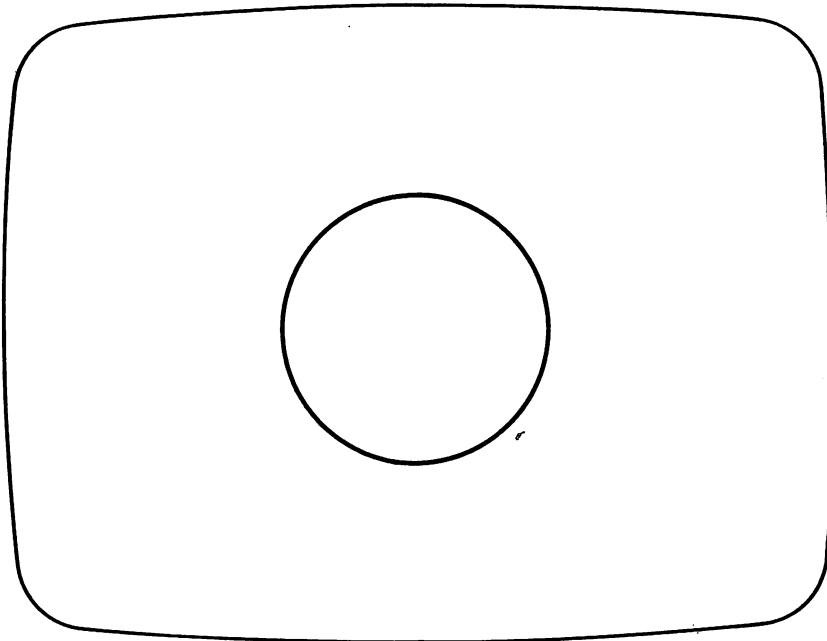


Figure 4-6. CIRCLE (100,100),50,1

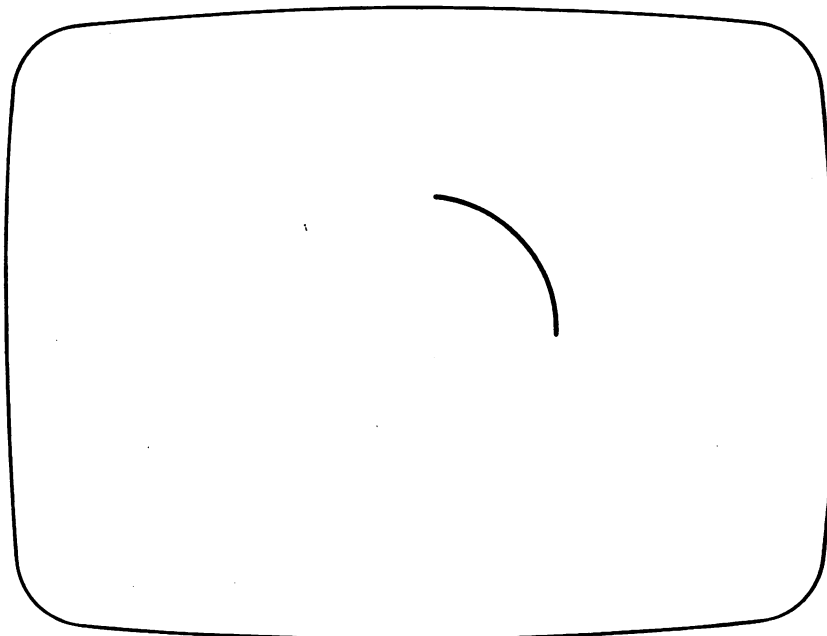


Figure 4-7. CIRCLE (100,100),50,1,.1,1.5

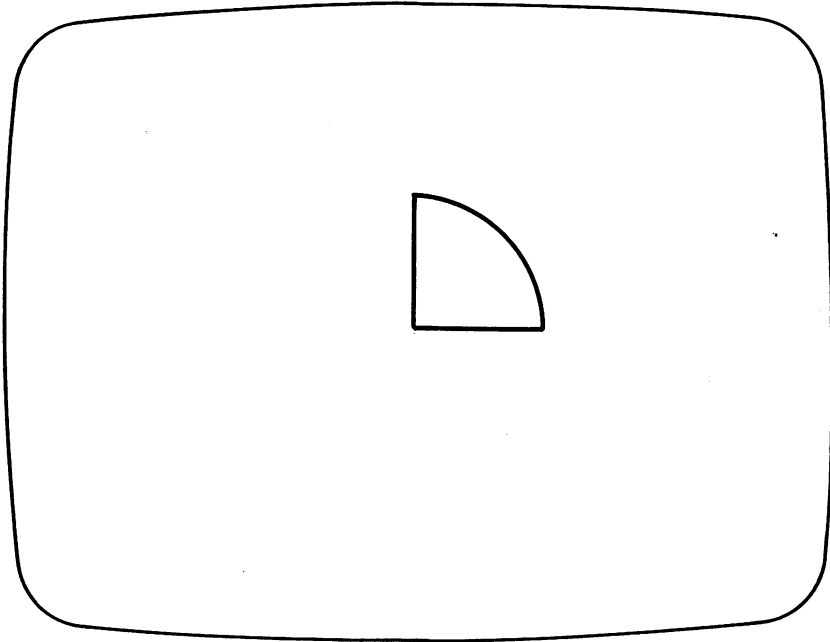


Figure 4-8. CIRCLE (100,100),50,1,-.1,-1.5

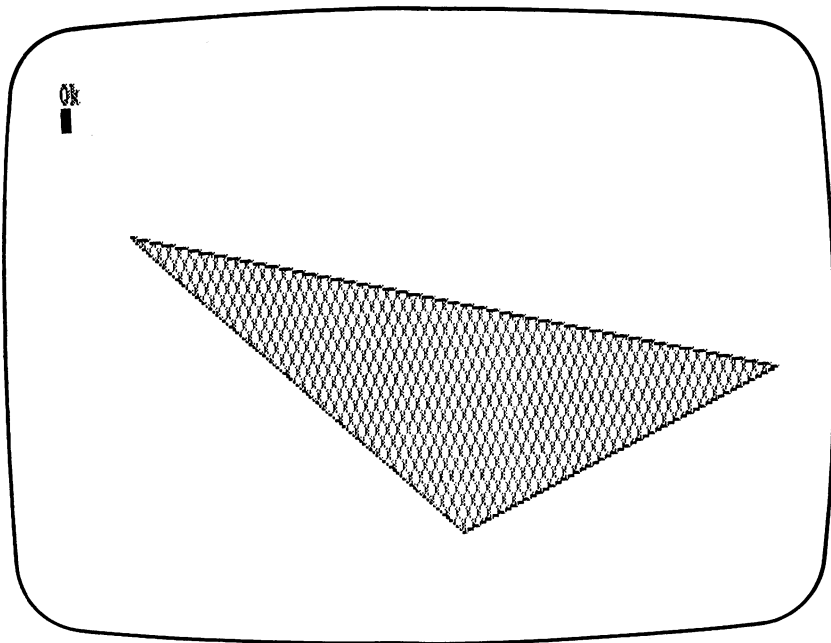


Figure 4-9a. A triangle painted in various styles.

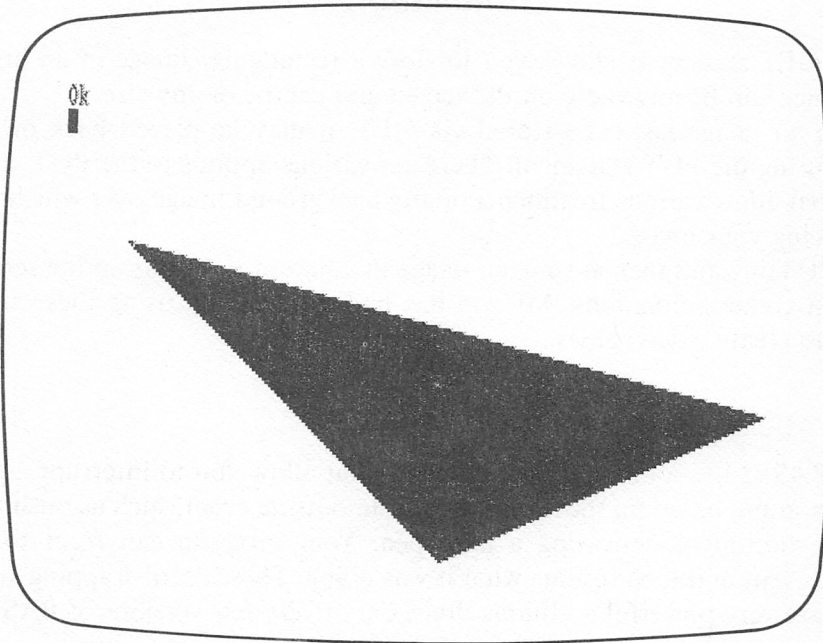


Figure 4-9b. A triangle painted in various styles.

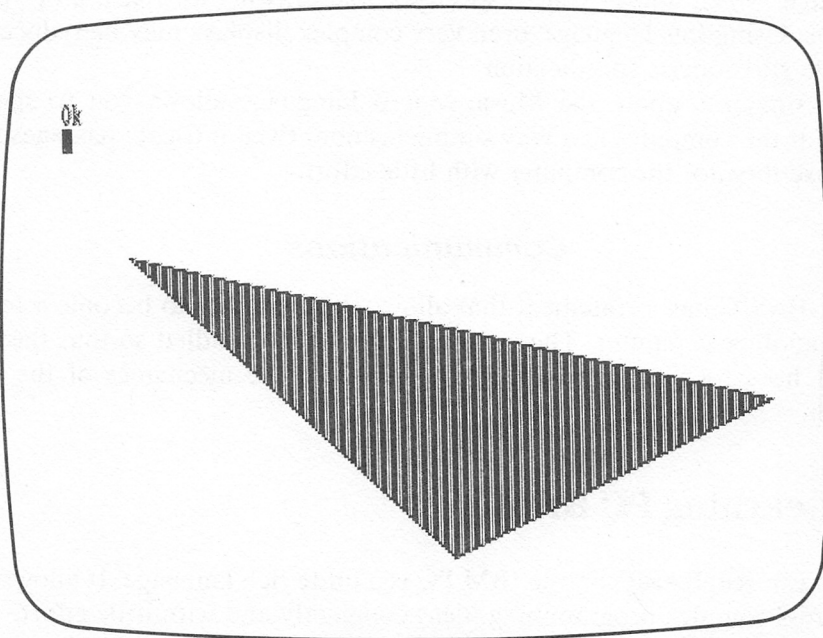


Figure 4-9c. A triangle painted in various styles.

Animation

The GET statement allows you to store a rectangular image in an array. The image can be anywhere on the screen and can be of any size.

Once an image has been stored via GET, it may be placed back on the screen using the PUT statement. There are various options to the PUT statement that allow various treatments of any background image over which you are placing your image.

By PUTting and then erasing an image in adjacent positions on the screen, you can create animations. My son has had a lot of fun using these statements to create video games.

Event Trapping

PC BASIC has a number of statements that allow you to interrupt a running program, based on the occurrence of an outside event, such as pushing a joystick button or activating a light pen. Your program can react to the outside event and then resume what it was doing. These event-trapping statements are very powerful additions that exist in very few versions of BASIC.

Graphics and Music Macro Languages

To aid in drawing complex shapes, PC BASIC includes a Graphics Macro Language, which allows you to specify a line drawing as a string of “pen” motions. Using this language, even very complex displays may be reduced to a simple and concise specification.

In a similar fashion, the Music Macro Language allows you to specify music for the computer in a very simple fashion. Even intricate passages may be transcribed for the computer with little effort.

Communications

PCjr BASIC has a statement that allows the computer to become a terminal of another computer. The communications are handled so that the user doesn't have to become too deeply involved in the mechanics of the data transfer.

4.3 Learning PC BASIC

As you can see, BASIC on the IBM PC is a quite rich language. It allows you to express complex programming ideas compactly and with little effort.

A language as rich as PC BASIC must be learned in stages, just as you would learn a foreign language such as French or German.

IBM has provided an introduction to PC BASIC for beginners in the book, *Hands on BASIC for Your IBM PCjr*. This tutorial comes with each machine and takes you step by step through the commonly used parts of PC BASIC. However, it covers only the most elementary portions of PC BASIC.

4.4 Summary of PCjr BASIC

ABBREVIATIONS: [] indicates optional item
 ... indicates preceding item may be repeated as many times as desired

A. BASIC COMMANDS (Immediate Mode)

AUTO	Generate line numbers automatically.	AUTO [.] [number] [, increment]
CHDIR	Change directories.	CHDIR [d:] path
CONT	Continue a program interrupted by Break.	CONT
DELETE	To delete program lines.	DELETE [.] [line1] [, line2]
EDIT	To display a line.	EDIT [.] [line]
FILES	To display names of files matching filespec. If filespec is omitted, names of all files on the current diskette are displayed.	FILES "[filespec]
KILL	Erase files matching given filespec.	KILL "[filespec]
LIST	List lines of a program.	LIST [line[-[line]]] [, filespec]
LLIST	List lines of a program on the printer.	LLIST [line[-[line]]]
LOAD	Load a program from cassette or diskette.	LOAD filespec [, R]
MERGE	Merge program file with current program.	MERGE filespec
MKDIR	Create a directory.	MKDIR path
NAME	Rename a file.	NAME filespec AS filespec
NEW	Erase current program from RAM.	NEW
RENUM	Renumber program lines.	RENUM [newnum] [, [oldnum] [, increment]]
RESET	Close all files and clear the system buffer.	RESET
RMDIR	Remove a directory.	RMDIR path

SAVE	Save current program.	SAVE "filespec" [,A] [,P]
SYSTEM	Return to DOS.	SYSTEM
TRON	Display numbers of lines executed. (Trace On)	TRON
TROFF	Disable the Trace function.	TROFF

B. PROGRAM CONTROL

END	End execution of current program.	END
FOR... NEXT	Repeat statements between FOR and NEXT as long as variable is less than or equal to y. The value of variable starts at x and with each repetition is increased by step.	FOR variable=x TO y [STEP z] ...NEXT [variable] [,variable]...
GOSUB... RETURN	Transfer execution to line. Upon reaching RETURN, go to line after one containing GOSUB.	GOSUB line ...RETURN
GOTO	Transfer control to indicated line.	GOTO line
IF... THEN... ELSE	If expr is true, then execute clause1 (or GOTO line); otherwise execute clause2.	IF expr THEN clause1 [ELSE clause2] IF expr GOTO line [ELSE clause2]
ON... GOSUB	Round the value of expr to an integer n. Go to the subroutine beginning at the nth listed line number.	ON expr GOSUB line [,line]...
ON... GOTO	Round the value of expression to an integer n. Transfer control to the nth listed line number.	ON expression GOTO line[,line]...
STOP	Stop program execution.	STOP
SYSTEM	Exit BASIC and return to DOS.	SYSTEM
WHILE... WEND	Execute the statements in between WEND WHILE and WEND until expr becomes false.	WHILE expr ...WEND

C. KEYBOARD AND SCREEN CONTROL (TEXT MODE)

CLS	Clear the screen.	CLS
COLOR	Set color.	COLOR [foreground] [,background] [,border]
CSRLIN	Returns the current line of cursor.	x = CSRLIN

LOCATE	Locate cursor at column x and row y.	LOCATE [row][,[col] [, [cursor][,[start] [, [stop]]]]]
INKEY\$	Returns character input at keyboard.	x\$=INKEY\$
INPUT	Reads data typed on keyboard.	INPUT[;][["prompt string";] variable variable [,variable]
INPUT\$	Return a string of n characters read from the keyboard.	x\$=INPUT\$(n)
KEY	Set or display Function Keys.	KEY n,x\$ KEY LIST KEY ON KEY OFF
KEY(n)	Activate/deactivate Function Key trapping.	KEY(n) ON KEY(n) OFF KEY(n) STOP
LINE INPUT	Input an entire line of data to a string variable x\$.	LINE INPUT [;][["prompt string";] x\$
ON KEY	Define trap line for function keys or cursor control key.	ON KEY(n) GOSUB line
PCOPY	Copy contents of one screen page onto another	PCOPY oldpage,newpage
POS	Returns the horizontal position of the cursor.	POS(1)
PRINT	Display data on screen.	PRINT list of expressions
PRINT USING	Display formatted numerical data on screen.	PRINT #filenumber [USING x\$;] list of expressions
SCREEN	Return the ASCII code (z=1) or color (z=0) of character at (row,col).	SCREEN (row,col [,z])
SPC	Insert n spaces before printing.	PRINT SPC(n)
TAB	Move to column n before printing.	PRINT TAB(n)
WIDTH	Define number of characters across screen.	WIDTH size
WRITE	Display data on screen with commas between data items and quotation marks around strings.	WRITE list of expressions

D. FILE MANIPULATION

CLOSE	Close a file.	CLOSE [#][filenumber]
-------	---------------	-----------------------

CVI, CVS, CVD	Convert string N\$\$ to integer(I), single-precision(S), or double-precision(D) number.	x=CVI(N\$) x=CVS(N\$) x=CVD(N\$)
EOF	Returns true(-1) if currently at the end of file, false(0) otherwise.	EOF(filename)
GET	Read a record from a random file into a buffer.	GET [#]filename [,recordnumber]
INPUT #	To read data items from a sequential device or file and assign them to program variables.	INPUT #filename, variable [,variable] ...
LINE- INPUT #	Input an entire line from a file.	LINEINPUT #filename, stringvar
LSET	Move data to a random-access file buffer, left-justified.	LSET stringvar = x\$
MKIS\$, MKSS\$, MKD\$	Convert numerical values to strings to write to a random access file.	x\$=MKIS\$(integer) x\$=MKSS\$(s.precision) x\$=MKD\$(d.precision)
OPEN	Open a file.	OPEN filespec [FOR mode] AS #filename [LEN=recordlength]
PRINT #	Write data to a sequential file.	PRINT #filename, list of expressions
PRINT # USING	Write formatted numerical data to a sequential file.	PRINT #filename [USING x\$;] list of expressions
PUT	Write a record in a buffer for a random access file.	PUT [#]filename [,recordnumber]
RSET	Move data to a random-access file buffer, right justified.	RSET stringvar = x\$
WRITE #	Write data to a sequential file. Insert commas between data items and delimit strings with quotation marks.	WRITE #filename, list of expressions

E. VARIABLE AND CONSTANT MANIPULATION

CDBL CINT CSNG	Convert a number x to double-precision (CDBL), integer (CINT), or single-precision. (CSNG)	CDBL(x) CINT(x) CSNG(x)
DATA	Identifies data items read by READ statements.	DATA constant [,constant]...

DEFINT	Define a variable to be of integer type, single-precision type, double-precision type, or string type.	DEFTYPE letter[-letter] [,letter[-letter]]...
DEFSNG		
DEFDBL		
DEFSTR		
LET	Assign values to variables.	[LET] variable=expression
READ	Assign values to variables using data items in DATA statements.	READ variable [,variable]...
RESTORE	Restore DATA pointer to specified line.	RESTORE [line]
SWAP	Interchange the values of variables variable1 and variable2.	SWAP variable1, variable2

F. ARRAYS

CLEAR	Set all numeric variables to 0, all string variables to null; define the end of user memory and amount of stack space.	CLEAR [[,bytesmemory] [stackspace]]
DIM	Define size of an array.	DIM var(subscripts) [,var(subscripts)] ...
ERASE	Eliminate arrays from a program.	ERASE arrayname [,arrayname]
OPTION BASE	Define the initial subscript value (0 or 1).	OPTION BASE n

G. GRAPHICS AND SOUND

BEEP	Activate the speaker.	BEEP
CIRCLE	Draw a circle, ellipse, or sector.	CIRCLE (xcntr,ycntr), radius [,color] [,start,end[,aspect]]
COLOR	Define current background color and palette.	COLOR [bkgrnd] [,palette]
DRAW	To draw a figure specified by the given string.	DRAW string
GET	Store pixel colors or a specified rectangle in an array.	GET (x1,y1)-(x2,y2), arrayname
LINE	Draw a straight line or rectangle.	LINE [(x1,y1)]-(x2,y2) [,color][,B[F], style]]
NOISE	Produce noise	source, volume, duration
ON PLAY	Allows continuous music during a program.	ON PLAY(note) GOSUB

PAINT	Color a region of the screen.	PAINT (x,y) [[,paint [,boundary [,background]]]] line
PALETTE	Control hardware palette	PALETTE Open [attribute] [,color]
PALETTE USING	Set colors of current palette	PALETTE USING [array name (starting)]
PLAY	Play music as coded in string.	PLAY string
POINT	Returns the color (0-3) of the pixel (x,y).	POINT(x,y)
PMAP	Converts an expression to physical coordinates.	PMAP(exp,n)
PSET	Sets the pixel (x,y) to the specified color.	PSET (x,y) [,color]
PRESET	Sets the pixel (x,y) to the background color. (Color 0)	PRESET (x,y)
PUT	Display the image stored in array at the pixel (x,y).	PUT (x1,y1), arrayname[,action]
SCREEN	Enable/disable color/graphics.	SCREEN [mode] [,[,burst][[,lapage] [,vpage]]]
SOUND	Generate tones on the speaker.	SOUND frequency, duration
VIEW	Define a viewport within the screen	VIEW [[SCREEN] [(x1,y1)- (x2,y2) [,[,color][, [,boundary]]]]]
WINDOW	Redefine coordinates of the screen	WINDOW [SCREEN] (x1,y1)- (x2,y2)

H. COMMUNICATIONS

CLOSE	Close a communications file.	CLOSE [#]filename
COM	Enable/disable trapping of input from communications adapter n.	COM(n) ON COM(n) OFF COM(n) STOP
ON COM(n)	Define trap line for input from communications adapter n.	ON COM(n) GOSUB line
OPEN 'COM...'	Open communications file filename associated with communications adapter n, with given speed (baud rate), parity, transmit/receive data bits, stop bits.	OPEN "COMn:[speed] [,parity][,data] [,stop]" AS filename
TERM	Emulate a terminal.	TERM

I. ARITHMETIC FUNCTIONS

ABS	Return the absolute value of X.	ABS(X)
ATN	Return the arctangent of X.	ATN(X)
COS	Return the cosine of X.	COS(X)
DEF FN	Define user function.	DEF FNname(X)= expression in X
EXP	Return the exponential of X.	EXP(X)
FIX	Returns the integer part of X.	FIX(X)
INT	Returns the largest integer less than or equal to x.	INT(x)
RAN- DOMIZE	Choose a seed for the random number generator.	RANDOMIZE
SGN	Returns the sign of X.	SGN(X)
SIN	Returns the sine of X.	SIN(X)
SQR	Returns the square root of X.	SQR(X)
TAN	Returns the tangent of X.	TAN(X)
TIMER	Returns the number of seconds since midnight or since system reset.	TIMER
VAL	Returns the value of the number determined by the initial segment of the string x\$.	VAL(x\$)

J. STRING FUNCTIONS

ASC	Return the ASCII code of the first ASC(x\$)	character of x\$
CHR\$	Return the character having ASCII code n.	CHR\$(n)
DAT\$	Return the date in the form mm-dd-yyyy.	DAT\$
INSTR	Return the position of the first instance of the string x\$ in the string y\$, beginning the search at character n.	INSTR(x\$,n [,m])
LEFT\$	Returns the leftmost n characters of x\$.	LEFT\$(x\$,n)
LEN	Returns the length of x\$.	LEN(x\$)
HEX\$	Return a string which is the hexadecimal value of the decimal n.	HEX\$(n)

MID\$	Returns a string of length m beginning with the nth character of x\$	MID\$(x\$,n,m)
OCT\$	Returns a string which is the octal value of the decimal n.	OCT\$(n)
RIGHT\$	Returns the n rightmost characters of x\$.	RIGHT\$(x\$,n)
SPACE\$	Returns a string of n spaces.	SPACE\$(n)
STR\$	Converts the number x into a string.	STR\$(x)
STRING\$	Returns a string of length n copies of the character ASCII m (first form) or n copies of the first character of x\$ (second form).	STRING\$(n,m) STRING\$(n,x\$)
TIMES	Returns the time as a string hh:mm:ss .	TIMES

K. EXTERNAL DEVICES

INP	Returns the byte read from port n.	INP(n)
ON PEN	Defines trap line for light pen input.	ON PEN GOSUB line
ON STRIG	Defines trap line for joystick button n (=0 or 2).	ON STRIG GOSUB line
OUT	Transmits a byte to port n.	OUT(n)
PEN	Enable/disable light pen trapping.	PEN ON PEN OFF PEN STOP
PEN(n)	Read light pen coordinates, service routine n.	PEN(n)
STICK	Return the x-coordinate (n=0 or 1) or y-coordinate (n=2 or 3) of joystick A or B.	STICK(n)
STRIG	Enables trapping of joystick buttons.	STRIG ON STRIG OFF
STRIG(n)	Returns the status of joystick buttons.	STRIG(n)
WAIT	Await input from port.	WAIT port,n[,m]

L. ERROR TRAPPING

ERR	Returns the error code of the most recent error.	ERR
-----	--	-----

ERL	Returns the program line number which generated the most recent error.	ERL
ERROR	Simulates the occurrence of error n.	ERROR n
ON ERROR	On error, transfer control to line.	ON ERROR GOTO line
RESUME	Clear error condition and transfer control to line after error line (NEXT), to the error line ([0]) or to line.	RESUME [0] RESUME NEXT RESUME line

M. MACHINE LANGUAGE INTERFACE

BLOAD	Load a binary file from diskette into memory beginning at offset.	BLOAD filespec [,offset]
BSAVE	Save a memory image beginning at offset and continuing for length bytes.	BSAVE filespec, offset,length
CALL	Call a machine language subroutine.	CALL numericvariable [(variable[,variable] ...)]
DEF SEG	Define the current segment of memory.	DEF SEG [=address]
DEFUSR	Define the starting address of a machine language subroutine.	DEF USR[digit]=offset
PEEK	Returns the contents of the indicated memory location.	PEEK(address)
POKE	Stores the indicated value in the specified memory location.	POKE address,value
USR	Call user-defined machine language subroutine n with argument x.	USR[digit](x)
VARPTR	Return the offset address of the indicated variable or file control block.	VARPTR(variable) VARPTR(#filenumber)

N. MISCELLANEOUS STATEMENTS AND FUNCTIONS

CHAIN	Pass control to another program.	CHAIN [MERGE] filespec [,line] [,ALL] [,DELETE range]]]
COMMON	Specify variables to be passed to a CHAIN'ed program.	COMMON variable [,variable]...
MOTOR	Turn cassette motor on/off.	MOTOR state

ON TIMER	Transfers control after a given period of time.	ON TIMER(n) GOSUB line
REM	Remark. Rest of line ignored.	REM remark ' remark

O. PRINTER

LLIST	List program lines on the printer.	LLIST [line[-line]]
LPRINT	Print constants and values of variables.	LPRINT [list of expressions]
LPRINT USING	Print formatted numerical data.	LPRINT USING x\$; list of expressions[;]

4.5 What is DOS?

If you attach a disk drive to your PCjr system, you will also require a piece of software to control the traffic between the disks and the rest of the system. Such a piece of software is called a **disk operating system**, or **DOS** for short. The particular disk operating system available for the PCjr is a product of Microsoft, Inc. and is variously known as MS-DOS or PC-DOS. Actually, there have been several versions of MS-DOS introduced at various times during the last few years. In order to run the disk drives of the PCjr, it was necessary to bring out yet another version of MS-DOS, this one called **DOS 2.10**.

You may purchase a copy of DOS 2.10 from your computer dealer. Note, however, that DOS 2.10 is always an optional item. Even if you purchase an enhanced system, with the disk drive already installed, it will be necessary to purchase DOS 2.10.

DOS 2.10 consists of two DOS diskettes (the main DOS diskette and the supplementary DOS diskette) and a reference manual containing a summary of the various DOS commands. In addition, you may purchase the **DOS Technical Reference Manual**, although you are unlikely to need this advanced manual unless you are a professional programmer.

To use DOS 2.10, first insert the main DOS diskette into the disk drive. (Actually, you'll be using a copy. More about making copies of diskettes later.) Next, turn on the monitor and the system unit. The system unit has circuitry that detects the presence of a diskette in the drive. This circuitry reads the main portion of the DOS program into RAM and begins to run it.

DOS then acts as the computer "manager." It takes care of the "house-keeping chores" such as handling communications between the various system components. You may communicate with DOS via a set of commands. For example, you may tell DOS to start BASIC by typing the command:

BASIC

followed by pressing the ENTER key. In response to this command, DOS will turn over control of the computer to the BASIC program cartridge you have inserted in one of the cartridge slots. You may then type in a BASIC program, recall a BASIC program from diskette, run a BASIC program, or perform any other BASIC activity. You may terminate BASIC by typing the BASIC command:

SYSTEM

and pressing the ENTER key. This command returns control of the computer to DOS.

DOS 2.10 is quite a powerful operating system. In fact, the big brothers of the PCjr, the IBM PC and PC/XT, may also use DOS 2.10. On these larger machines, DOS 2.10 can be used to manage multiple disk drives and, possibly, even one or more hard disks.

Chapters 9 and 10 give a brief introduction to DOS. We will cover the above sequence of actions in greater detail and will give you a clearer idea of the various commands DOS can perform.

4.6 IBM Software and Publications

In this section, we will summarize the IBM software and Publications available for the PCjr. Note, however, that this list is based on what is available as of this writing. In the months ahead, many new products for the PCjr will certainly be released by both IBM and other manufacturers. So take this list as only a sample of what is to come.

Entertainment

Program Name	Adventure in Serenia
Version	1.00
Description	Fantasy adventure
Medium on Which Supplied	Diskette
Comments	PCjr colors may be different than colors produced on the PC or PC/XT
Program Name	Animation Creation
Version	1.00
Description	Tool to assist in creating animations in BASIC by drawing frames (as in filming a movie).
Medium on Which Supplied	Diskette

Program Name	Crossfire
Version	1.00
Description	Arcade-style game. Defend city streets from a swarm of insects.
Medium on Which Supplied	Cartridge
Requirements	Cartridge BASIC
Program Name	Casino Games
Version	1.05
Description	Assorted games of chance. Good for parties.
Medium on Which Supplied	Diskette
Program Name	Juggles' Butterfly
Version	1.00
Description	
Medium on Which Supplied	Diskette
Requirements	Cartridge BASIC
Program Name	Mine Shaft
Version	1.00
Description	Arcade-style game. Looking for diamonds in a dangerous mine shaft.
Medium on Which Supplied	Cartridge
Requirements	Cartridge BASIC
Program Name	Mouser
Version	1.00
Description	Arcade-style game. Trap mice in a farmhouse.
Medium on Which Supplied	Cartridge
Program Name	Scuba Venture
Version	1.00
Description	Arcade-style game. Search for sunken treasure.
Medium on Which Supplied	Cartridge
Program Name	Strategy Games
Version	1.00
Description	Assorted games.
Medium on Which Supplied	Diskette

Education

Program Name Adventures in Math
Version 1.00
Description Game to improve basic math skills
Medium on Which Supplied Diskette
Requirements cartidge BASIC
Comments PCjr does not allow you to set the foreground and background colors.

Program Name Arithmetic Games, Set 1
Version 1.00
Description Teaches elementary arithmetic via games
Medium on Which Supplied Diskette
Requirements DOS 2.10, Cartridge BASIC
Comments Color present only on RGB monitor (includes IBM Color Display)

Program Name Arithmetic Games, Set 2
Version 1.00
Description Teaches elementary arithmetic via games
Medium on Which Supplied Diskette
Requirements DOS 2.10, Cartridge BASIC
Comments Color present only on RGB monitor (includes IBM Color Display)

Program Name Bumble Games
Version 1.00
Description Teaches use of number pairs to describe positions.
Medium on Which Supplied Diskette
Requirements Cartridge BASIC

Program Name Bumble Plot
Version 1.00
Description Teaches young children point plotting concepts
Medium on Which Supplied Diskette
Requirements Cartridge BASIC

Program Name LOGO
Version 1.00
Description Popular graphics language
Medium on Which Supplied Diskette
Requirements DOS 2.10

Program Name	Monster Math
Version	1.00
Description	Arithmetic drill at six different levels
Medium on Which Supplied	Diskette
Requirements	Cartridge BASIC

Program Name	Turtle Power
Version	1.00
Description	Turtle graphics
Medium on Which Supplied	Diskette
Requirements	None

Finance

Program Name	Dow Jones Reporter
Version	1.00
Description	
Medium on Which Supplied	Diskette
Requirements	DOS 2.10, Cartridge BASIC

Program Name	Home Budget, jr
Version	1.00
Description	Organize your personal finances on the PCjr
Medium on Which Supplied	Diskette
Requirements	DOS 2.10, Cartridge BASIC

Program Name	Multiplan
Version	1.10
Description	Spreadsheet
Medium on Which Supplied	Diskette
Requirements	DOS 2.10

Program Name	VisiCalc
Version	1.20
Description	Spreadsheet
Medium on Which Supplied	Diskette
Requirements	DOS 2.10

Word Processing

Program Name	EasyWriter
Version	1.15
Description	Word processor commonly used with PC and PC/XT
Medium on Which Supplied	Diskette
Requirements	DOS 2.10, 80 character wide screen

Program Name	HomeWord
Version	1.00
Description	Simplified word processor designed specifically for home use.
Medium on Which Supplied	Diskette
Requirements	None
Program Name	PeachText
Version	1.00
Description	Word processor in common use on PC and PC/XT
Medium on Which Supplied	Diskette
Requirements	DOS 2.10, 80 character wide screen
Program Name	Word Proof
Version	1.00
Description	Spelling Checker
Medium on Which Supplied	Diskette
Requirements	DOS 2.10

Time and Data Management

Program Name	pfs:FILE
Version	1.05
Description	Data filing system
Medium on Which Supplied	Diskette
Requirements	DOS 2.10
Program Name	pfs:REPORT
Version	1.05
Description	File search and report generator
Medium on Which Supplied	Diskette
Requirements	DOS 2.10
Program Name	Time Manager
Version	1.05
Description	Computerized appointment calendar
Medium on Which Supplied	Diskette
Requirements	DOS 2.10

Programming

Program Name	BASIC Compiler
Version	1.00
Description	Compiles PC BASIC programs
Medium on Which Supplied	Diskette
Requirements	DOS 2.10, LINK
Comments	Must have sufficient memory
Program Name	BASIC Programming Development System
Version	1.05
Description	Translates structured BASIC into PC BASIC, produces formatted listings and cross-reference listings
Medium on Which Supplied	Diskette
Requirements	DOS 2.10
Program Name	Macro Assembler
Version	1.00
Description	Assembles 8088 source code
Medium on Which Supplied	Diskette
Requirements	Dos 2.10
Program Name	Personal Editor
Version	1.00
Description	Program editor
Medium on Which Supplied	Diskette
Requirements	DOS 2.10, 80 character screen width
Program Name	Professional Editor
Version	1.00
Description	Program editor with advanced features
Medium on Which Supplied	Diskette
Requirements	DOS 2.10, 80 character screen width

Communications

Program Name	Personal Communications Manager
Version	1.00
Description	Manages modem communications
Medium on Which Supplied	Diskette
Requirements	Modem
Comments	

In this section, we describe the various IBM publications on the PC jr.

IBM PCjr Guide to Operations—This is essential to getting started. It contains set-up instructions as well as instructions for operating the diagnostic routines. Supplied with the PCjr.

BASIC Reference Manual—The fundamental sourcebook for information about PC BASIC. If you plan to do any BASIC programming, you will be spending a great deal of time with this manual. Supplied with Cartridge BASIC. Also available separately.

BASIC Made Easy for the IBM PCjr—Tutorial on BASIC for young children. Must be purchased separately.

IBM Personal Computer Disk Operating System User's Guide (Level 2.10)—A manual for the user of applications programs running under DOS 2.10. Contains a DOS tutorial as well as a summary of the elementary DOS commands, the line editor EDLIN and the debugger DEBUG. Supplied with DOS 2.10.

IBM Personal Computer Disk Operating System Technical Reference (Level 2.10)—A manual for the DOS programmer. Concerned mainly with advanced aspects of DOS. Must be purchased separately.

Hands-On BASIC for the IBM PCjr—An introduction to BASIC programming. Supplied with each system.

IBM PCjr Technical Reference Manual—Gives details of the architecture of the PCjr, technical specifications of all PCjr components. Must be purchased separately.

IBM PCjr Hardware, Maintenance, and Service Manual—For service technicians. Must be purchased separately.

LOGO: Programming with Turtle Graphics—An introduction to the IBM LOGO graphics language. Must be purchased separately.

4.7 IBM SAMPLER AND TUTORIALS

IBM has designed the PCjr so that you may learn much of what you need to know from various tutorials supplied with the machine.

Keyboard Adventure—This tutorial instructs you on the features of the keyboard. It is stored in ROM in all PCjr systems.

Exploring the PCjr—A self-paced, diskette-based tutorial supplied with the enhanced unit. Introduces the user to the PCjr, the function keys, system unit, DOS, BASIC, diskettes and diskette drives, and printers.

PCjr Sampler Diskette—A collection of useful Programs supplied with the enhanced unit, including:

SHOPPING LIST

TYPEWRITER

MONTHLY EXPENSES

HOME LOANS

PHONE CONNECTOR

BRAIN TEASERS

GAMES

MORTGAGE

This sample allows you to put the PCjr to work immediately, without buying any initial software.

5

TYPICAL SYSTEM CONFIGURATIONS

In this chapter, we describe four typical PC Junior systems, ranging from a simple starter system to a deluxe “all-inclusive” system. In each case, we describe the components you will need. Since the transformer comes with the system unit, we have not listed it as a separate component.

The software summary chart (page 57-62) can help you to determine which programs are suitable for a particular system.

5.1 A Starter System

Our starter system is the least expensive PCjr system. Here are the components:

- System Unit - Basic 64k unit
- Display - Any user-supplied television
PCjr connector for TV
- Keyboard - Battery operated
- Cassette - A user-supplied cassette recorder
Cassette adapter cable
- BASIC - Cassette BASIC
- Joystick - PCjr joystick

This system is illustrated in Figure 5-1.

The starter system may be used for entertainment, education, communications, and do-it-yourself programming. However, note that screen displays in the starter system are 40 characters across.

Long-term program and data storage is limited to cassette. Moreover, this system lacks a printer.

All of IBM's games can be played from the keyboard. The joystick, which may be omitted, only makes the games more arcade-like.

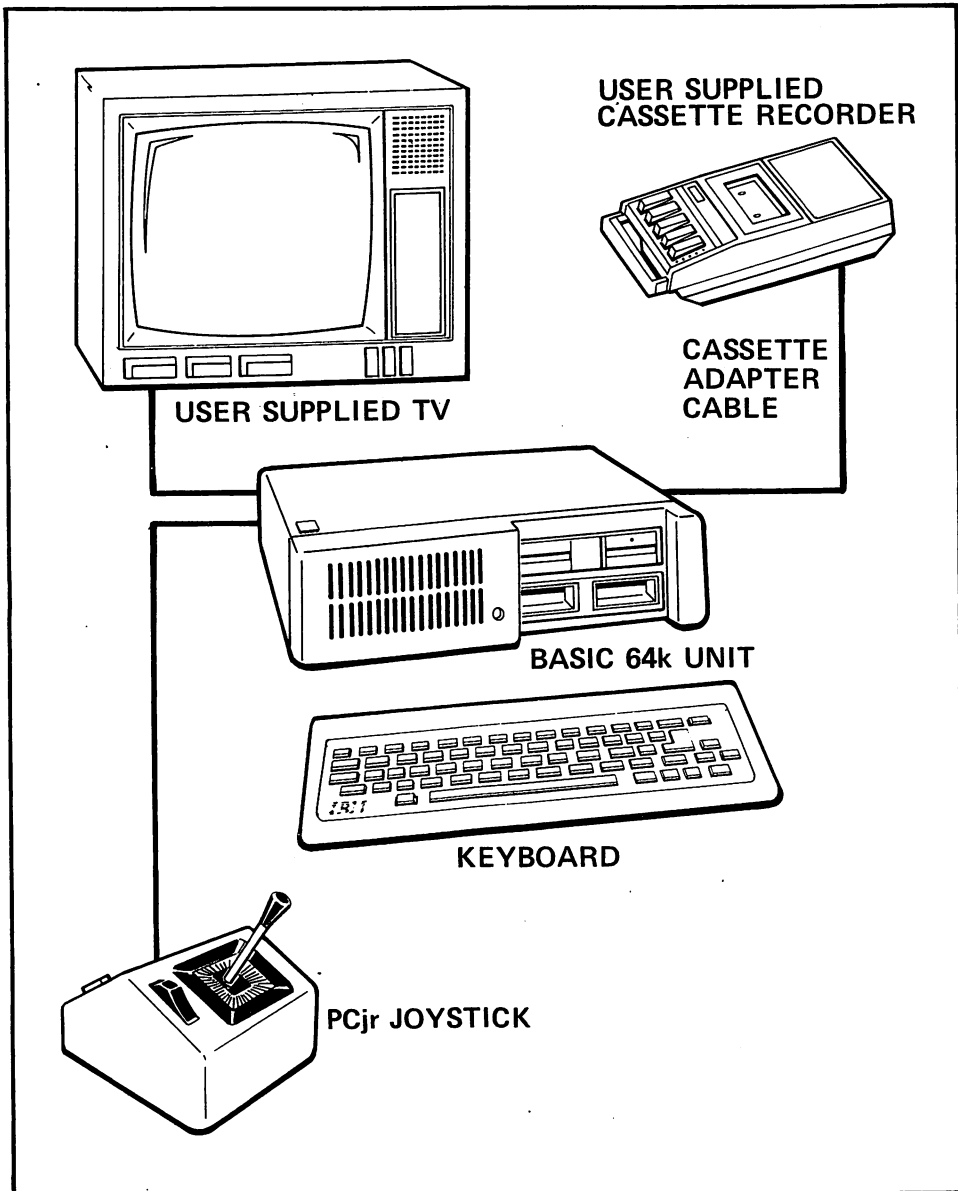


Figure 5-1. A PCjr starter system.

The starter system has two expansion slots free.

5.2 An Upgraded Starter System

The next system adds a printer and a monitor to the starter system. Here are the components:

System Unit	- Basic 64k unit
Display	- Black and white monitor Monitor cable User-supplied color TV set for games PCjr connector for TV
Display adapter	- 64k Memory and Display Adapter
Keyboard	- Battery operated
Cassette	- A user-supplied cassette recorder PCjr adapter cable for cassette
BASIC	- Cassette BASIC
Joystick	- PCjr joystick
Printer	- PC Compact Printer

This system is illustrated in Figure 5-2.

The monitor offers several advantages. In text mode, it allows you to choose between a 40-character or 80-character line width. In graphics mode, it offers higher resolution than is possible on a television set. Moreover, the picture will tend to be sharper on a monitor than on a television set. These advantages are balanced by the lack of color.

A black and white monitor can be purchased for \$150-\$200. To connect a black and white monitor to your system, you will need a monitor cable (non-IBM). Check with your dealer for the proper cable.

The 64k Memory Expansion and Display Adapter option provides an additional 64k of memory to the system, for a total of 128k. The display adapter provides the connection for plugging a monitor into your system and allows for 80 column wide display. The 64k Memory Expansion and Display Adapter board uses one of the expansion slots, so the upgraded starter system has only one expansion slot free for further system expansion.

5.3 A Minimal Diskette System

Our next system is the smallest system on which you can do word processing. It includes a diskette drive and supplements Cassette with the more powerful Cartridge BASIC.

System Unit	- Extended 128k Unit with these options: Diskette Drive 64k Memory and Display Expansion
Display	- Black and white monitor Monitor cable
Keyboard	- Battery operated

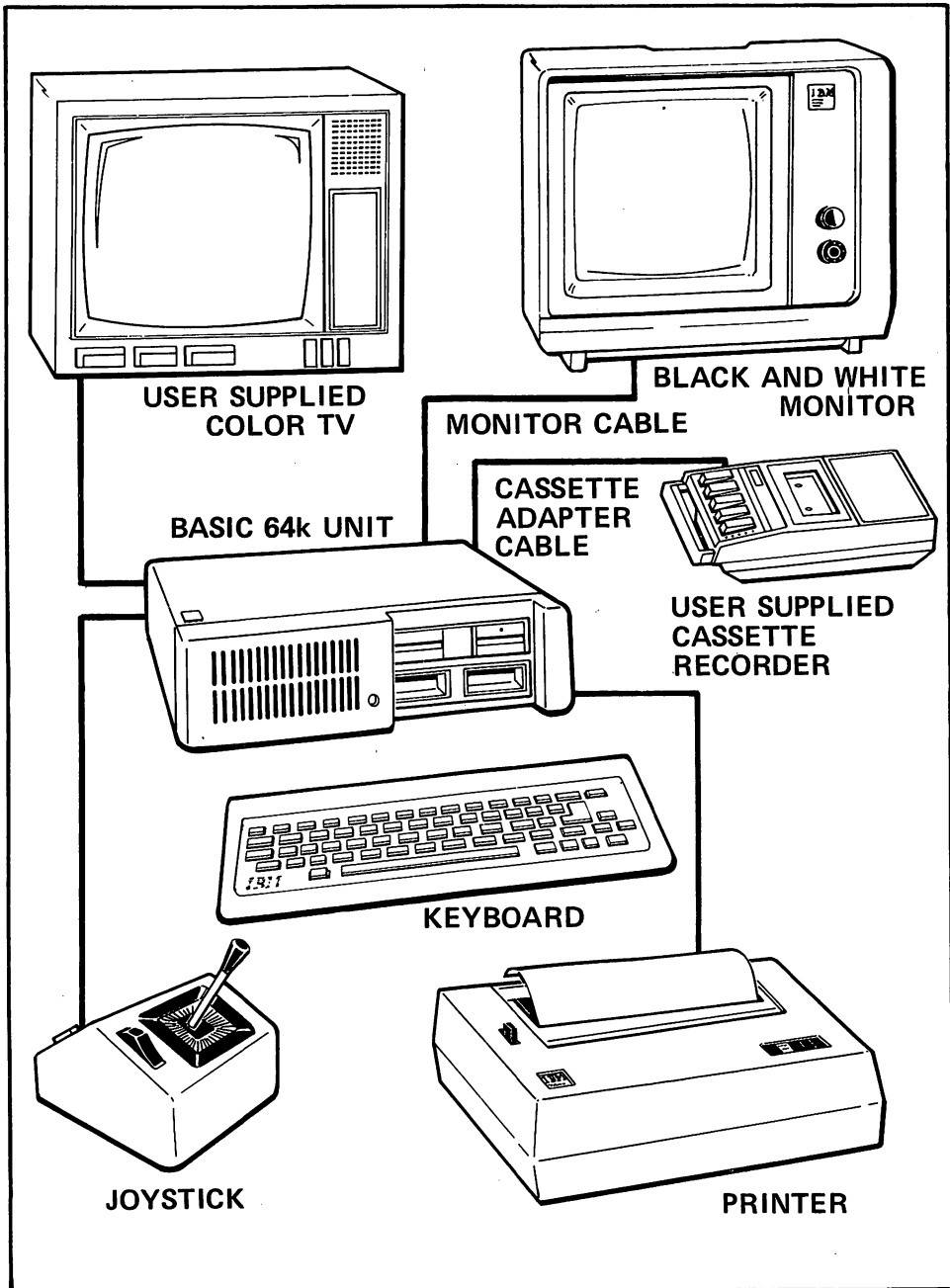


Figure 5-2. An upgraded starter system.

- BASIC - Cartridge BASIC
- Printer - PC Compact Printer
- Operating system - DOS 2.10 Disk Operating System

The Minimal Diskette System has a single free expansion slot and is shown in Figure 5-3.

5.4 A Deluxe System

The Deluxe System is an upgraded version of the Minimal Diskette System. There are three differences:

1. The black and white monitor is replaced with the IBM Color Display.
2. The IBM Compact Printer is replaced with the IBM Graphics Printer.
3. A modem has been added.

The IBM Color Display allows you to have the best of both worlds: graphics in color and an 80 character wide display for word processing. The IBM Graphics Printer is faster and more versatile than the IBM Compact Printer. In addition, it prints on standard sheets of continuous paper, which can result in a considerable saving over the thermal paper required by the IBM Compact Printer.

The modem allows you to communicate over telephone lines with other computers and outside information sources.

Here is the list of components in the Deluxe System:

- System Unit - Extended 128k Unit with these options:
 - Diskette Drive
 - 64k Memory and Display Expansion
- Display - IBM Color Display
 - PCjr Adapter Cable for Color Display
- Keyboard - Battery operated
- BASIC - Cartridge BASIC
- Printer - IBM Graphics Printer
 - PCjr Parallel Printer Attachment
- Operating system - DOS 2.10 Disk Operating System
- Modem - PCjr Internal Modem

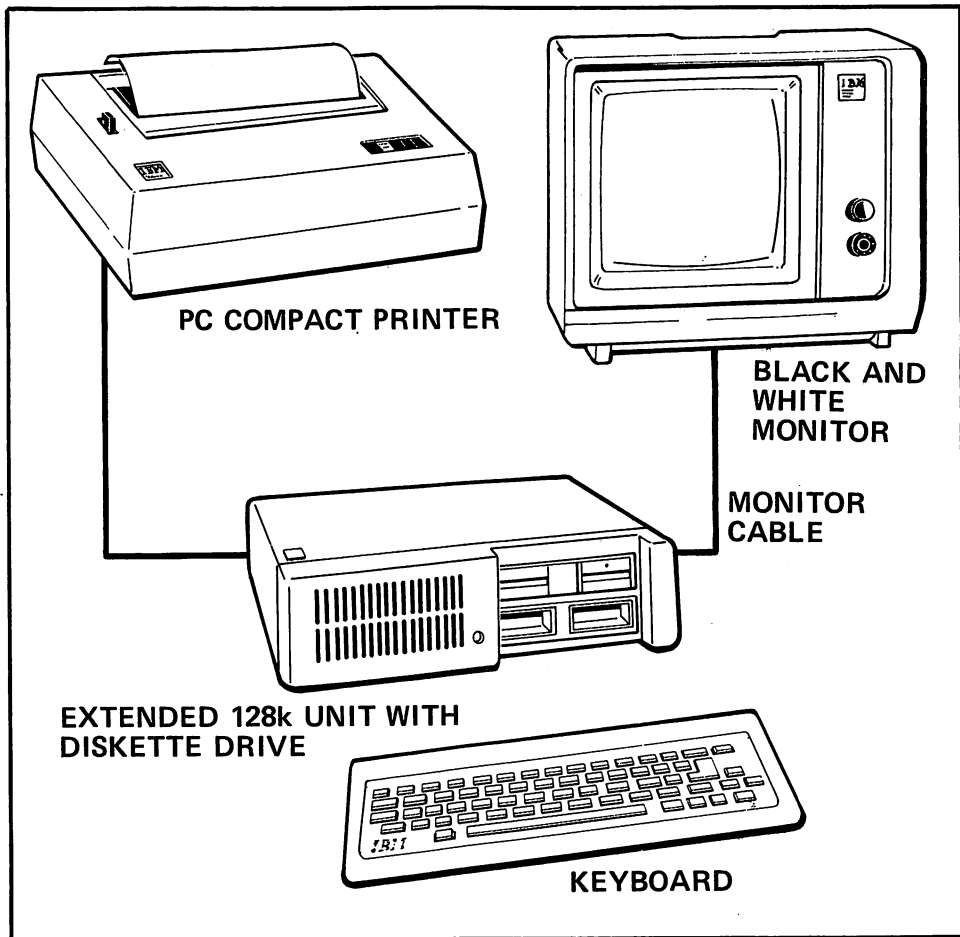


Figure 5-3. A minimal diskette system.

The Deluxe System has no free expansion slots. This system is illustrated in Figure 5-4.

5.5 What Will It All Cost?

To help you select a system, we have priced out the four systems considered in the preceding sections using the suggested retail prices established by IBM at the time of PCjr's introduction. Each price estimate is on an order form giving the various hardware options and their IBM serial numbers. In reviewing these costs, note that we did not include any software other than Cartridge BASIC and DOS 2.1, where appropriate. To arrive at a final cost for your system, you should include both software and other miscellaneous costs, such as any computer furniture you plan to buy.

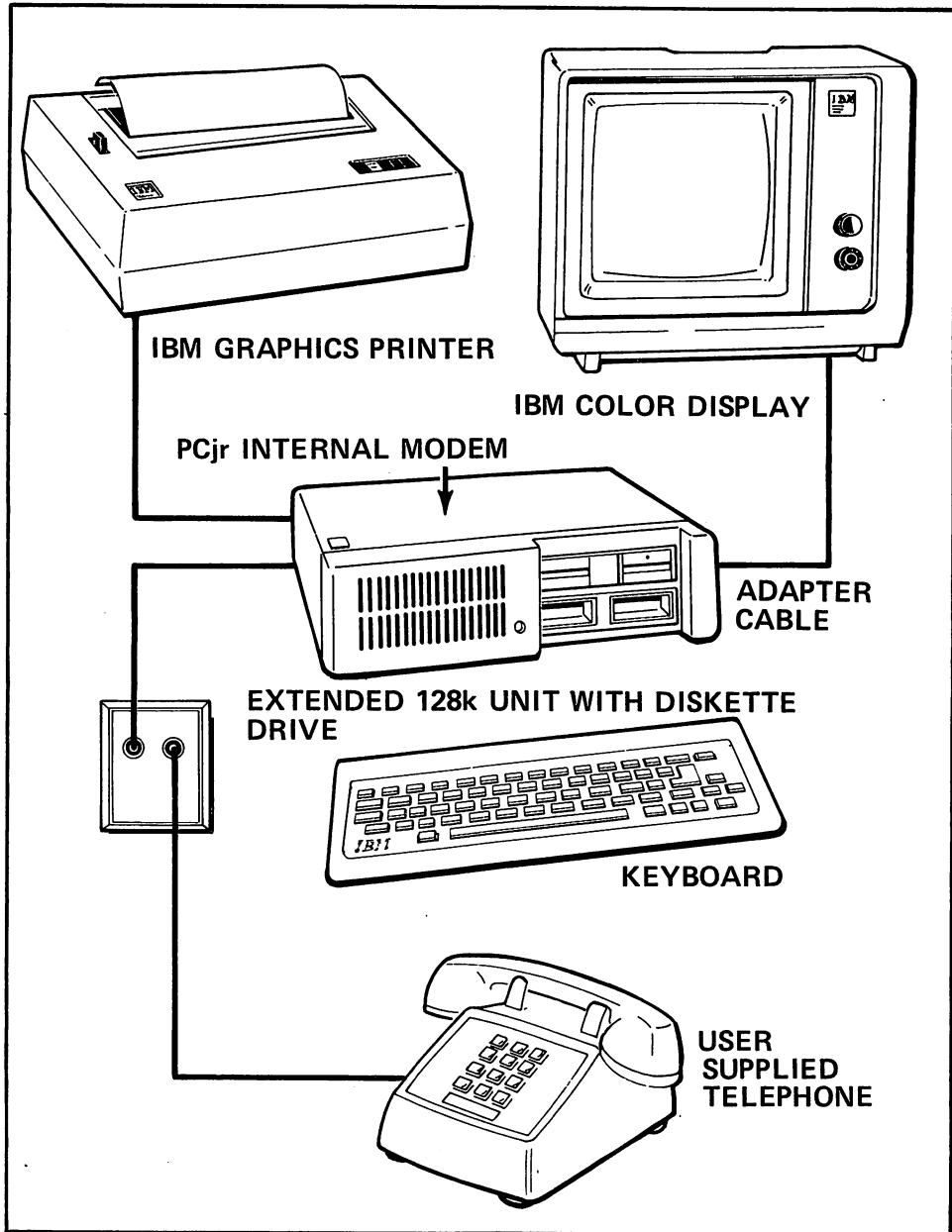


Figure 5-4. A deluxe system.

Cost Estimate: PCjr Starter System

<i>Component Name</i>	<i>Serial Number</i>	<i>Price</i>	
PCjr Basic Model	8600004	669.00	X
PCjr Enhanced Model	8600067	1269.00	
PCjr Diskette Drive*	8600005	480.00	
64k Memory and Display Expansion*	8600007	140.00	
PCjr Internal Modem	8600008	199.00	
Parallel Printer Attachment	8600009	99.00	
Joystick	8600010	40.00	X
PCjr Keyboard Cable	8600012	20.00	
TV Connector	8600020	30.00	X
IBM Color Display	5152002	595.00	
Adapter Cable for IBM Color Display	8600021	20.00	
Adapter Cable for Cassette	8600022	30.00	X
Adapter Cable for Serial Devices	8600026	25.00	
IBM Compact Printer	5181001	175.00	
IBM Graphics Printer	5153001	680.00	
PCjr BASIC Reference Manual with Cartridge	6024101	75.00	
DOS 2.1	6024120	65.00	
Black and white monitor with cable		approx. 200.00	
TOTAL HARDWARE COST		\$769.00	

* Included with Enhanced Model.

Cost Estimate: PCjr Upgraded Starter System

<i>Component Name</i>	<i>Serial Number</i>	<i>Price</i>	
PCjr Basic Model	8600004	669.00	X
PCjr Enhanced Model	8600067	1269.00	
PCjr Diskette Drive*	8600005	480.00	
64k Memory and Display Expansion*	8600007	140.00	X
PCjr Internal Modem	8600008	199.00	
Parallel Printer Attachment	8600009	99.00	
Joystick	8600010	40.00	X
PCjr Keyboard Cable	8600012	20.00	
TV Connector	8600020	30.00	X
IBM Color Display	5152002	595.00	
Adapter Cable for IBM Color Display	8600021	20.00	
Adapter Cable for Cassette	8600022	30.00	X
Adapter Cable for Serial Devices	8600026	25.00	
IBM Compact Printer	5181001	199.00	X
IBM Graphics Printer	5153001	680.00	
PCjr BASIC Reference Manual with Cartridge	6024101	75.00	
DOS 2.1	6024120	65.00	
Black and white monitor with cable		approx. 200.00	X
TOTAL HARDWARE COST		\$1278.00	

* Included with Enhanced Model.

Cost Estimate: PCjr Minimal Diskette System

<i>Component Name</i>	<i>Serial Number</i>	<i>Price</i>	
PCjr Basic Model	8600004	669.00	_____
PCjr Enhanced Model	8600067	1269.00	X _____
PCjr Diskette Drive*	8600005	480.00	_____
64k Memory and Display Expansion*	8600007	140.00	_____
PCjr Internal Modem	8600008	199.00	_____
Parallel Printer Attachment	8600009	99.00	_____
Joystick	8600010	40.00	X _____
PCjr Keyboard Cable	8600012	20.00	_____
TV Connector	8600020	30.00	X _____
IBM Color Display	5152002	595.00	_____
Adapter Cable for IBM Color Display	8600021	20.00	_____
Adapter Cable for Cassette	8600022	30.00	_____
Adapter Cable for Serial Devices	8600026	25.00	_____
IBM Compact Printer	5181001	175.00	X _____
IBM Graphics Printer	5153001	680.00	_____
PCjr BASIC Reference Manual with Cartridge	6024101	75.00	X _____
DOS 2.1	6024120	65.00	X _____
Black and white monitor with cable		approx. 200.00	X _____
TOTAL HARDWARE COST		\$1854.00	

* Included with Enhanced Model.

Cost Estimate: PCjr Deluxe System

Component Name	Serial Number	Price	
PCjr Basic Model	8600004	669.00	_____
PCjr Enhanced Model	8600067	1269.00	_____ X _____
PCjr Diskette Drive*	8600005	480.00	_____
64k Memory and Display Expansion*	8600007	140.00	_____
PCjr Internal Modem	8600008	199.00	_____ X _____
Parallel Printer Attachment	8600009	99.00	_____ X _____
Joystick	8600010	40.00	_____ X _____
PCjr Keyboard Cable	8600012	20.00	_____
TV Connector	8600020	30.00	_____
IBM Color Display	5152002	595.00	_____ X _____
Adapter Cable for IBM Color Display	8600021	20.00	_____ X _____
Adapter Cable for Cassette	8600022	30.00	_____
Adapter Cable for Serial Devices	8600026	25.00	_____
IBM Compact Printer	5181001	175.00	_____
IBM Graphics Printer	5153001	680.00	_____ X _____
PCjr BASIC Reference Manual with Cartridge	6024101	75.00	_____ X _____
DOS 2.1	6024120	65.00	_____ X _____
Black and white monitor with cable		approx. 200.00	_____
TOTAL HARDWARE COST		\$3042.00	

* Included with Enhanced Model.

6

OTHER MEMBERS OF THE IBM FAMILY

If you are considering buying a PCjr, you are probably also thinking about the other members of the IBM family. You may have a PC or PC/XT at work that you want to communicate with. Also, if you are shopping for a fully-equipped PCjr, you will find that it is not that much more money to upgrade to a PC system. In this chapter, we will point out the similarities and differences among the PCjr, PC, and PC/XT systems.

6.1 The IBM PC and IBM PC/XT

The IBM PC and PC/XT have much in common with the PCjr. All three computers use the same microprocessor (the INTEL 8088¹), process information at the same speed², and can use the DOS 2.10 operating system (although at this time PC and PC/XT users are using either DOS 1.1 or DOS 2.00).

Here, however, are some differences.

Memory—The PC comes with 64k of memory, the PC/XT with 128k. However, each can be equipped with as much as 576k. The amount of RAM in your system becomes significant if you wish to run some of the modern integrated software programs, such as 1-2-3.³ Such programs often require a minimum of 256k. The memory of the PC and PC/XT is **parity checked**, whereas that of the PCjr is not. (Parity checking is a way of alerting the computer to possible loss or alteration of data in RAM.)

¹Intel 8088 is a trademark of Intel Corporation.

²PCjr throughput may be slower due to choice of disk drives and screen memory refresh requirements.

³Registered trademark of Lotus Development Corp.



Figure 6-1. IBM PC.

Diskette Drives and Hard Disks—The PC and PC/XT have space for two disk drives in their respective system units. Actually, if you use halfheight drives like the ones in PCjr, the PC system unit could accommodate four drives. The PC/XT comes with a 10-megabyte hard disk and one floppy disk drive as standard items, leaving no room for disk expansion in the system unit.

Monitors—The PC and PC/XT can use the same monitors as the PCjr. In addition, however, they can use the IBM Monochrome Display, which features exceptionally crisp letters and is very good for word processing.

Expansion Slots—The PC has five expansion slots, and the PC/XT has eight. Thus, each of them has much greater expansion flexibility than the PCjr.

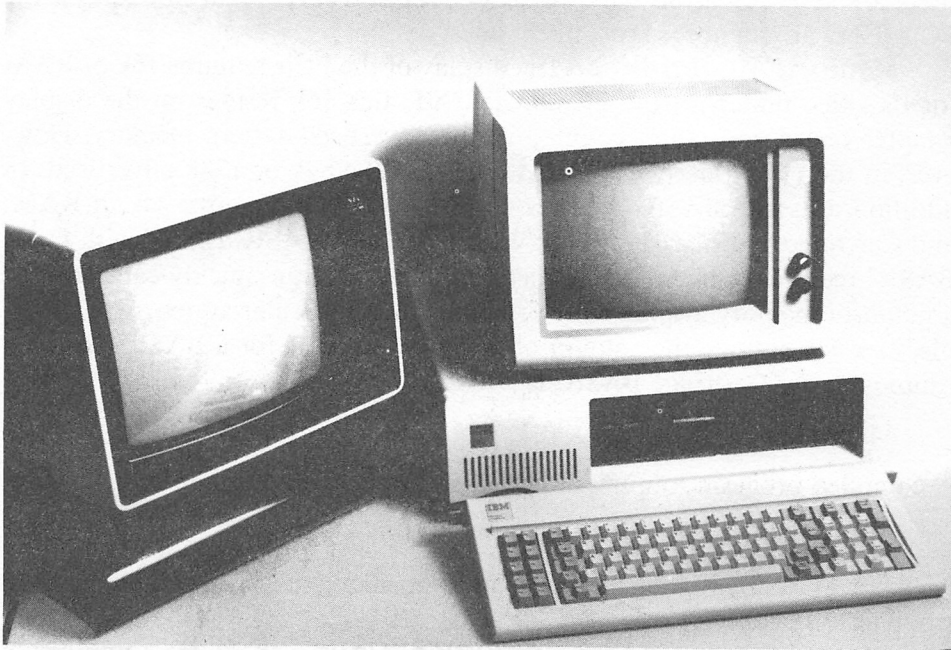


Figure 6-2. IBM PC/XT.

BASIC Language—PCjr BASIC runs only on the PCjr. It is more powerful than Advanced BASIC for the PC and PC/XT. (It includes all Advanced BASIC commands and some others as well.) The areas in which PCjr has statements unavailable in Advanced BASIC are as follows:

Music—Music Macro Language allowing control of three-voice music synthesizer chip.

Graphics—Enhanced palette control

Communications—Terminal emulation mode

Multi-Page Display—Copy one screen page to another

6.2 Software Compatibility

IBM introduced its first personal computer in late 1981. Within two years, there were more than 3,000 programs available for it. Many of these programs will run as is on the PCjr. However, others will need to be changed in order to run on the PCjr, while still others are incompatible with the PCjr. In this section, we will attempt to shed some light on the question of compatible software.

PCjr Cartridge BASIC is a superset of PC Advanced BASIC. So, in principle, any program written in PC BASIC should work on the PCjr. However,

there are several reasons why PC or PC/XT software might not run on the PCjr. Here are the three most likely ones:

Memory Availability—The display of the PCjr requires 16k of RAM for its video buffer. In the PC and PC/XT, this 16k resides on the display adapter board and is not counted in the advertised system memory. However, in the PCjr, this 16k of RAM is part of the 64k or 128k advertised. In addition, Cassette BASIC requires 4k of RAM, DOS requires 24k of RAM, and Cartridge BASIC requires 6k. When Cartridge BASIC is used with DOS, BASIC requires only 3k. Using these numbers, we can quickly calculate the amount of memory available for programs in a particular system. For example, in a 64k system, the amount of memory available for a BASIC program running under Cartridge BASIC is

$$64k - 16k - 6k = 42k .$$

A compiled program running on the same system could use up to

$$64k - 16k = 48k .$$

A PC or PC/XT program that uses more memory than is available will not run on the PCjr.

Direct Memory Access and BIOS Calls—There are a number of instructions in BASIC that access memory directly. Examples are PEEK, POKE, DEFSEG, BSAVE, and BLOAD. Some memory addresses in the PCjr are different from those of the PC and PC/XT. So be wary of programs using direct memory access. Similarly, be careful of programs that use BIOS calls rather than DOS calls. Be especially careful with copy-protected programs. Some copy protection schemes may not work on the PCjr and will not allow the program to run.

Screen Width—Many programs require a screen width of 80 characters. Such programs will not run unless the PCjr system has a Memory Expansion and Display Adapter board installed.

7

PCjr SYSTEM PLANNING AIDS

7.1 Planning a PCjr System

Well, now we've supplied all the raw information. It's time for you to go to work designing your PCjr system. Here's how to proceed.

1. Review the applications we have presented and decide which are most important to you. Decide on the program(s) you would most like to use.
2. What are the hardware implications of your chosen applications programs? (If you want to do communications, you'll need a modem. Games are best played on a color monitor. Word processing requires an 80 character wide screen.)
3. On the basis of your hardware requirements, decide what components you need.
4. Consult the following order form for the available components and their prices.

7.2 System Component Summary and Price List

The following pages contain a detailed summary of the various components available from IBM, their serial numbers, and IBM's initial suggested retail prices.

SUGGESTED RETAIL PRICES FOR PCjr HARDWARE

<i>Component Name</i>	<i>Serial Number</i>	<i>Price</i>	
PCjr Basic Model	8600004	669.00	_____
PCjr Enhanced Model	8600067	1269.00	_____
PCjr Diskette Drive*	8600005	480.00	_____
64k Memory and Display Expansion*	8600007	140.00	_____
PCjr Internal Modem	8600008	199.00	_____
Parallel Printer Attachment	8600009	99.00	_____
Joystick	8600010	40.00	_____
PCjr Keyboard Cable	8600012	20.00	_____
TV Connector	8600020	30.00	_____
IBM Color Display	5152002	595.00	_____
Adapter Cable for IBM Color Display	8600021	20.00	_____
Adapter Cable for Cassette	8600022	30.00	_____
Adapter Cable for Serial Devices	8600026	25.00	_____
IBM Compact Printer	5181001	175.00	_____
IBM Graphics Printer	5153001	680.00	_____
Black and white monitor with cable		approx. 200.00	_____

* Included with Enhanced Model.

SUGGESTED RETAIL PRICES FOR PCjr SOFTWARE

<i>Software</i>	<i>Serial Number</i>	<i>Price</i>	
<i>SYSTEM SOFTWARE</i>			
PCjr BASIC Reference Manual			
with Cartridge	6024101	75.00	_____
DOS 2.1	6024120	65.00	_____
<i>ARCADE STYLE GAMES</i>			
Mouser	6024087	35.00	_____
Scubaventure	6024088	35.00	_____
Crossfire	6024091	35.00	_____
Mineshaft	6024092	35.00	_____
<i>SOFTWARE ON DISKETTE</i>			
Monster Math	6024072	30.00	_____
Animation Creation	6024082	30.00	_____
Home Word	6024090	35.00	_____
Bumble Games	6024094	40.00	_____
Juggles' Butterfly	6024095	35.00	_____
Bumble Plot	6024096	40.00	_____
Personal Communications Manager	6024100	100.00	_____
Turtle Power	6024109	50.00	_____
Adventures in Math	6024112	35.00	_____
Home Budget, Jr.	6024130	45.00	_____

SUGGESTED RETAIL PRICES FOR PCjr PUBLICATIONS AND MISCELLANEOUS COMPONENTS

<i>Item</i>	<i>Serial Number</i>	<i>Price</i>	
Guide to Operations*	1502292	21.25	_____
Hands-On BASIC*	1502290	17.50	_____
BASIC Reference Manual (without cartridge)	1504702	11.25	_____
Technical Reference Manual	1502293	35.00	_____
Hardware Maintenance & Service Manual	1502294	80.00	_____
DOS 2.1 Technical Reference Manual	6024125	30.00	_____
BASIC Made Easy/ PCjr	6024116	13.00	_____
LOGO : Programming With Turtle Graphics	1502229	24.00	_____
Keyboard Overlays	8600013	10.00	_____
PCjr Carrying Case	8600023	60.00	_____

*Included with system unit.

USER GUIDE TO DOS 2.1

8

USING DOS FOR THE FIRST TIME

Most PCjr users will be using diskettes rather than cassettes to store their programs and data. However, to use the diskette drives, you must put your computer under the control of the program DOS 2.1. In this chapter and the next, we will present a brief introduction to this important topic. Since this part of the book is a tutorial, we have included **TEST YOUR UNDERSTANDING** questions to quiz you on your understanding of the material covered. The answers to these questions are at the end of each section. In addition, there are exercises, with the answers at the back of the book.

8.1 On Diskettes and Diskette Drives

Your floppy diskette drives are a critical part of your computer system. They allow you to store and retrieve both programs and data. Before we proceed any further, let's get acquainted with these remarkable devices.

The Anatomy of a Diskette

To store information, the diskette drives use 5¼-inch floppy diskettes. Diskettes come in single-sided and double-sided versions. The single-sided diskettes may be written on only one side, a double-sided diskette on both sides. The diskette drive of your PCjr uses double-sided diskettes that can accommodate roughly 360,000 characters (about 100 double-spaced typed pages).

Figure 8-1 shows the essential parts of a diskette. The diskette itself is a magnetically coated circular piece of mylar plastic that rotates freely within a stiff jacket. The jacket is designed to protect the diskette. The interior of the jacket contains a lubricant that helps the diskette rotate easily. The diskette is

sealed inside. You should *never* attempt to open the protective jacket. The labels on the jacket identify the contents of the diskette.

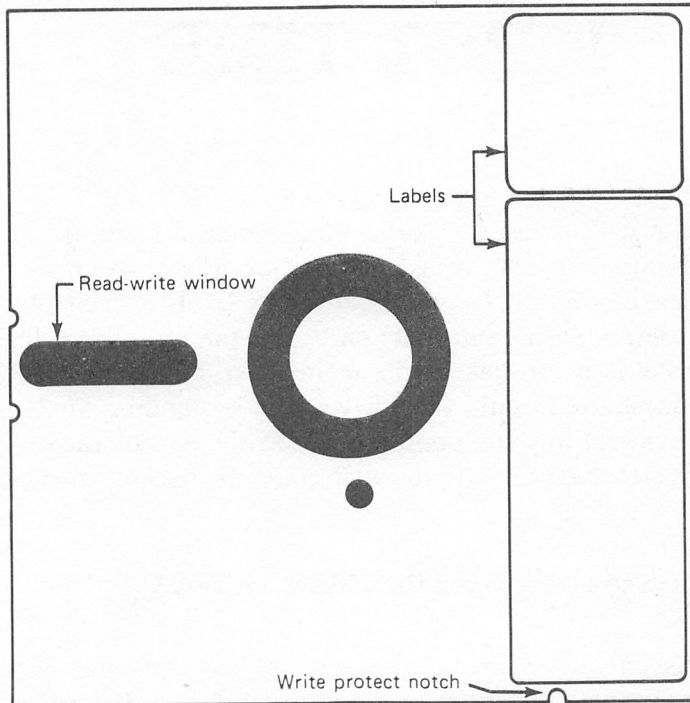


Figure 8-1. A diskette.

The diskette drive reads and writes on the diskette through the **read-write window**. **NEVER**, under any circumstances, touch the surface of the diskette. Diskettes are very fragile. A small piece of dust or even oil from a fingerprint could damage the diskette and render parts of the information on it totally useless.

The **write protect notch** allows you to prevent changes to information on the diskette. When this notch is covered with one of the metallic labels provided with the diskettes, the computer may read the diskette, but it will not write or change any information on the diskette. To write on a diskette, the write protect notch must be uncovered.

You should have a few blank diskettes on hand. Take a moment to inspect one of them and locate the parts of the diskette described above. (The labeling on your diskettes may differ from that shown in Figure 8-1.)

Cautions in Handling Diskettes

Diskettes are extremely sensitive. Here are some tips in using them.

1. Always keep a diskette in its paper envelope when it is not in use.
2. Store diskettes in a vertical position just like you would a phonograph record.
3. Never touch the surface of a diskette or try to wipe the surface of a diskette with a rag, handkerchief, or other piece of cloth.
4. Keep diskettes away from extreme heat, such as radiators, direct sun, or other sources of heat.
5. Never bend a diskette.
6. When writing on a diskette label, use only a felt-tipped pen. Never use any sort of instrument with a sharp point.
7. Keep diskettes away from magnetic fields, such as those generated by electrical motors, radios, TVs, tape recorders, phones, and other electrical devices. A strong magnetic field may erase data on a diskette.
8. Never remove a diskette while the drive is running. (You can tell if a drive is running by the sound of the motor and the “in use” light on the front of the drive.) Doing so may cause permanent damage to the diskette.

The above list of precautions may seem overwhelming to someone starting out. However, once you set up a suitable set of procedures for handling and storing diskettes, you’ll find that they are a reliable, long-lasting storage medium.

Using Diskettes

To insert a diskette into a diskette drive, open the door of the drive. Turn the diskette so that the label side is facing up and the read-write window is closest to the computer. Gently push the diskette all the way into the drive. Close the drive door. The diskette may now be read by the computer. To remove a diskette from a drive, first be sure that the light to the left of the drive door is off. Lift the drive door and gently pull the diskette forward and out of the drive.

TEST YOUR UNDERSTANDING 1

Take a blank diskette and practice inserting it in the diskette drive on the left. Remove the diskette from the drive.

8.2 Starting Your PCjr

To control the flow of information to and from the diskette drive, we need a program called an **operating system**. Such a program acts as a manager for all the activities that go on in the computer. More specifically, it coordinates the flow of information among the keyboard, video display, RAM, ROM, diskette drive, and any other peripheral devices you may add to your computer system.

The official operating system of the IBM PC is called **IBM DOS** (IBM Disk Operating System, pronounced IBM dóss), also called **MS-DOS*** or **PC-DOS**, or just **DOS** for short. Actually, DOS has undergone several revisions since its original version. As of this moment, the latest version is DOS 2.10. And this version runs on all three IBM personal computers, the PCjr, the PC, and PC/XT.

When you purchased your system, you should have also bought a copy of the **Disk Operating System** manual. Just inside the rear cover of the DOS manual is a plastic jacket that contains two diskettes, one labeled DOS and the other **DOS Supplementary Programs**. The first diskette is your master copy of the programs necessary to operate your diskette drives. This diskette is extremely important. So important, in fact, that it does not have a write protect notch. This means that you can never write on this diskette. (No chance for accidentally altering its programs!) We shall refer to this diskette as the **master DOS diskette**.

To use DOS, it's first necessary to read DOS into the computer. Ordinarily, this would be done with a *copy* of the master DOS diskette, rather than with the original itself. However, on your first pass, you don't yet have any extra copies of the DOS diskette, so we must use the master. Here's the procedure to follow.

Starting Your Computer

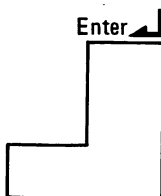
1. Insert the DOS diskette into the diskette drive. The label side should be up. Push the diskette all the way to the rear of the drive. Close the drive door.
2. Turn on your monitor.

*MS stands for Microsoft, the corporation that designed the operating system.

3. Turn on the printer (if one is connected).
4. On the system unit (the box in which the diskette drive sits) you will find the computer On-Off switch. (See Fig. 8-2.) Flick it to the up position. For about 30-45 seconds, the computer does diagnostic testing to determine if all its components are in good working order. When the tests are complete the computer should respond with a beep and the display:

Enter today's date (mm-dd-yy):

5. Type in today's date (in the format 4-22-99 for April 22, 1999). Press the **ENTER** key, which is the large key with the symbol:



The computer will respond with a display similar to:

Enter correct time (mm:hh:ss):

Type in the correct time in the format 14:03:00 for 2:03 PM. (The PC uses a 24-hour clock.)

The computer will respond with a display similar to:

```
The IBM Personal Computer DOS
Version 2.10 Copyright IBM Corp. 1981,1982,1983
A>_
```

The symbol A> is called the **DOS prompt** and it tells you that DOS is loaded and ready to accept commands.

6. In this chapter, we'll learn to use many of the DOS commands. At this point, let's give the command to read the BASIC programming language from the DOS diskette into RAM. Type:

```
basic
```

Press the **ENTER** key. The computer will respond with a display similar to:

```
The IBM Personal Computer Basic
Version 2.10 Copyright IBM Corp. 1981,1982,1983
xxxxx Bytes Free
Ok
-
```

Note the letters **Ok** in the last line of the display. These letters are the **BASIC prompt** and indicate that the computer language (called BASIC) is ready to accept instructions. The small blinking box is called the **cursor** and indicates the place on the screen where the next typed character will appear.

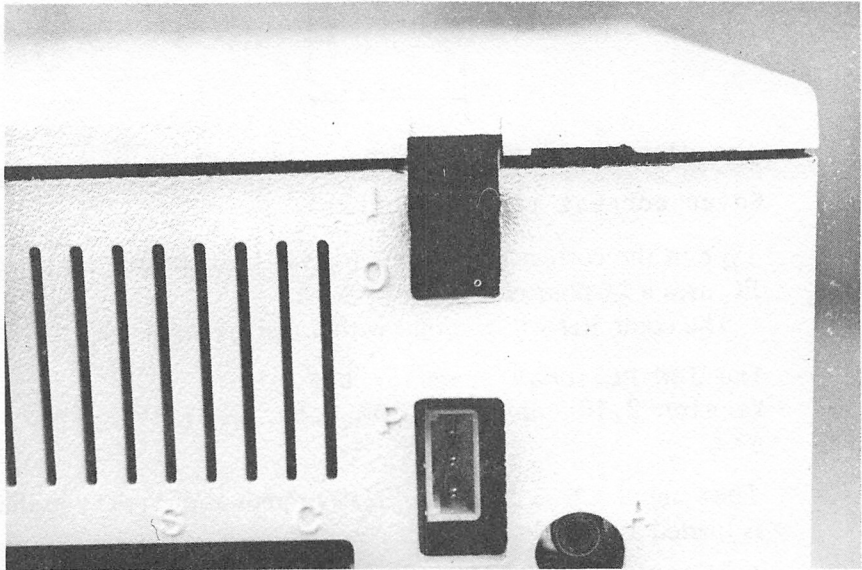


Figure 8-2.

The above instructions will load BASIC from diskette. If you wish to load Cartridge BASIC, just insert the cartridge into the cartridge slot on the left. Now type BASIC and press ENTER.

7. To return from BASIC to DOS, type:

`system`

and press ENTER. The computer will display the DOS prompt A>.

TEST YOUR UNDERSTANDING 1 (answers* on page 91)

- a. Turn on your computer and load DOS.
- b. Load BASIC.
- c. Return to DOS.

Turning Off the Computer

1. Turn off the system unit.
2. Turn off the monitor.
3. Remove any diskettes from the drives.

ANSWERS TO TEST YOUR UNDERSTANDING

1. (a) Follow the instructions 1-7 for Starting Your Computer.
- (b) Follow instruction 6.
- (c) Follow instruction 7.

8.3 More About Diskette Drives

The PCjr can be equipped with only a single diskette drive. However, DOS is capable of recognizing up to four drives. The diskette drives are given the names A:, B:, C:, and D: (note the colons). Drive names are used to refer to a drive within a command.

The single diskette drive on the PCjr can actually be referred to by either of the drive names A: or B:. At any given moment, only one of these names is in effect. You can tell which by looking at the DOS prompt. If the A: name is being used, then the prompt will be A>. If the B: name is being used, then the DOS prompt will be B>. You'll see below the reason for this rather odd use of names. The drive whose name is shown in the DOS prompt is called the **current drive** or the **default drive**. If you give a command without referring to a drive, then the current drive is assumed.

*Answers to TEST YOUR UNDERSTANDING questions (where appropriate) are given at the end of the current section.

To change the current drive:

1. Obtain the DOS prompt A> or B>.
2. Type the name of the new current drive (remember the colon). Press ENTER.

TEST YOUR UNDERSTANDING 1 (Answers on page 92)

- a. Turn on your computer and change the current drive from A: to B: .
- b. Change the current drive back to A: .

ANSWER TO TEST YOUR UNDERSTANDING

- 1: (a) After turning on the computer and obtaining the DOS prompt, type B: and press ENTER.
- (b) Type A: and press ENTER.

8.4 Backing Up Your DOS Diskette

Good programming practice dictates that you keep duplicate copies of all your diskettes. In computer language, a copy is called a **backup**. Making backups reduces the chance that you'll lose your programs and data due to accidents (power blackout, coffee spilled on a diskette, and so forth). It is an especially good idea to make a copy of the master DOS diskette the first time you use it. Later, you should use only the copy. The original DOS diskette should be stored in a safe place so that yet another copy can be made if the first copy is damaged. Here is the procedure for making a backup copy of a diskette. (The instructions are for single drive systems. For systems with two drives, see the comment below.)

Copying One Diskette Onto Another

We must copy the contents of the master DOS diskette onto a blank diskette.

1. Obtain the DOS prompt A>. Insert the DOS diskette into drive A:.
Type:

DISKCOPY

then press the **ENTER** key. The computer will give the display:

Insert source diskette in drive A:

Strike any key when ready

2. The source diskette, namely the DOS diskette, is already in drive A.; so strike any key. The computer will respond with the prompt:

Copying 9 sectors per track, 1 sides(s)

DOS will then copy a portion of DOS from the diskette into RAM. When it has copied as much as it can, it will display:

Insert target diskette in drive A
Strike any key when ready

Remove the DOS diskette and insert a blank diskette. Next, press any key. The computer will now copy the data in RAM onto the diskette. If there is more data to be copied from the DOS diskette, you will be directed to reinsert the DOS diskette. Then steps 1 and 2 will be repeated a number of times. After all the data has been copied, the computer will display the prompt:

Copy complete
Copy another? (Y/N)

Answer N to indicate that we do not wish to copy another diskette. The computer will now display the DOS prompt:

A>

3. Your blank diskette is now an exact copy of the original. At this point, you may give another DOS command or request BASIC.

TEST YOUR UNDERSTANDING 1 (Answer on page 94)

Make a copy of the DOS system diskette supplied with your diskette operating system.

From now on you should use only the **copy** of the master DOS diskette, and **not** the original. Put the original in a safe place, so that it can be used to make yet another copy if the current copy is damaged.

A WORD TO THE WISE

The backup procedure just described may be used to copy the contents of any diskette onto any other. Because diskettes are fragile, we strongly urge you to maintain duplicate copies of all your diskettes. A good procedure is to update your copies at the end of each session with the computer. This may seem like a big bother, but it will prevent untold grief if, by some mishap, a diskette with critical programs or data is erased or damaged.

TEST YOUR UNDERSTANDING 2 (Answer on page 94)

Use your copy of the master DOS diskette to make another copy. (A copy of a copy is just as good as the original!)

ANSWERS TO TEST YOUR UNDERSTANDING

1. Follow instructions 1-3 on page 92-93.
2. Follow instructions 1-3 on page 92-93, except start with the copy DOS diskette in drive A:.

8.5 The Keyboard

Let's examine the PCjr keyboard (see Fig. 8-3). This keyboard looks complex, but we'll cover the functions of the various keys slowly to give you time to feel comfortable with the design.



Figure 8-3. The PCjr keyboard.

The central section is very much like a typewriter keyboard. There are a few symbols that are not present on a typewriter, such as:

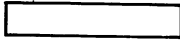
< > ^ ~ [] \ { }


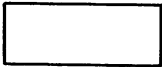
Also, you should note the following important differences from a typewriter keyboard:

1. There are separate keys for 1 (one) and l (el). (Many typewriters use the lowercase l to do double duty as a one.)

2. The number 0 (zero) has a slash through it. This is to distinguish it from the letter O.

Here are the functions of the other keys in the central portion of the keyboard:

 **Space bar.** Generates a blank space just like the space bar on a typewriter.

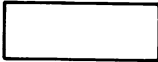
Shift 
 **Shift key.** Shifts keys to their uppercase versions. For keys with two symbols, the upper symbol takes effect. The uppercase versions are in effect only as long as the shift key is held down. Releasing the shift key causes keys to assume their lowercase meanings. Note that there are two shift keys, one on each side of the keyboard.

CapsLock



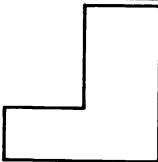
Caps Lock key. See the discussion on page 96.

Backspace



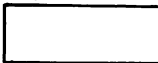
Backspace key. Moves the cursor back one space. Erases any letter it backs over.

Enter



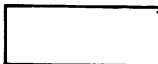
ENTER key. Similar to a carriage return key on a typewriter. Used to end a line and to place the cursor at the beginning of the next line. A line may be corrected with backspaces until ENTER is pressed.

Tab



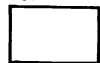
Tab key. Works like the tab key on a typewriter. Moves the cursor to the next tab stop.

Ctrl



Control key. Used in combination with other keys. For example, the key combination Ctrl-A means to simultaneously press Ctrl and A. Such combinations are used to generate control codes for screen and printer.

Esc



Escape key. Used to indicate that certain sequences of letters are to be interpreted as control codes.

Turn on your PC and obtain the DOS prompt A>. Strike a few keys to get the feel of the keyboard. Note that as you type, the corresponding characters will appear on the screen. Note, also, how the cursor travels along the typing line. It always sits at the location where the next typed character will appear.

As you type, you should notice the similarities between the IBM Personal Computer keyboard and that of a typewriter. However, you should also note the differences. At the end of a typewriter line you return the carriage, either manually or, on an electric typewriter, with a carriage return key. Of course, your screen has no carriage to return. However, you still must tell the computer that you are ready to move on to the next line. This is accomplished by hitting the **ENTER** key. If you depress the **ENTER** key, the cursor will then return to the next line and position itself at the extreme left side of the screen. The **ENTER** key also has another function. It signals the computer to accept the line just typed. Until you hit the **ENTER** key, you may add to the line, change portions, or even erase it. (We'll learn to do these editing procedures shortly.)

Keep typing until you are at the bottom of the screen. If you hit **ENTER**, the entire contents of the screen will move up by one line and the line at the top of the screen will disappear. This movement of lines on and off the screen is called **scrolling**.

As you may have already noticed, the computer will respond to some of your typed lines with error messages. Don't worry about these now. The computer has been taught to respond only to certain typed commands. If it encounters a command that it doesn't recognize, it will tell you so with an error message. It is extremely important for you to realize that these errors will in no way harm the computer. In fact, there is little you can do to hurt your computer (except by means of physical abuse, of course). Don't be intimidated by the occasional slaps on the wrist handed out by your computer. Whatever happens, don't let these "slaps" stop you from experimenting. The worst that can happen is that you might have to turn your computer off and start all over!

Use of the Caps Lock Key—In most computer work it is convenient to type using only capital letters. For one thing, capitals are larger and easier to read on the screen. You may turn off the lowercase letters by depressing the **Caps Lock** key. In this mode, the letter keys are automatically typed as capitals. Note, however, that the non-letter keys (such as **!** and **!**) still have two meanings. To type the upper symbol, you must still use the **SHIFT** key. To exit from the all-capitals mode, once again depress the **Caps Lock** key. With the **Caps Lock** key engaged, if you press the shift key and a letter key, a lowercase letter will be displayed.

System Reset

You may restart the computer from the keyboard by pressing the **Ctrl**, **Alt**, and **Del** keys simultaneously. This key sequence will return the computer to the state it was in just after being turned on. Both **RAM** and the screen will be erased.

TEST YOUR UNDERSTANDING 1 (Answers on page 97)

- a. Type your name on the screen.
- b. Erase the screen.
- c. Repeat a, using all capital letters. (Don't worry about the computer's response to your typing!)

Exercises

Type the following expressions on the screen. After each numbered exercise, clear the screen. (These exercises contain some typical expressions you will be typing when you use BASIC. They are provided as practice in typing and in manipulating the keyboard.)

- | | |
|----------------------|--|
| 1. 10 Print "Hello." | 7. 10 X=5: PRINT X |
| 2. 10 ARITH1=1.5378 | 8. 20 IF X>0 THEN 50 |
| 3. 10 PRINT 3+7 | 9. 10 LET X=10
20 LET Y=50.35 |
| 4. 20 LET A = 3-5 | 10. 200 Y = X*2 - 5
300 PRINT Y,"Y" |
| 5. 20 5% of 68 | |
| 6. 10 IF 38 > -5 | |

ANSWERS TO TEST YOUR UNDERSTANDING

1. (a) Type your name, ending the line with ENTER.
(b) Hit Ctrl and Home simultaneously.
(c) Hit Caps Lock. Now repeat part a.

9

AN INTRODUCTION TO DOS

DOS is involved in every aspect of PC use. It is no exaggeration to say that every time you sit down at your computer, you are using DOS. In this chapter, we will learn to use some of the most essential DOS commands.

9.1 Files and File Names

The contents of a diskette are broken into units called **files**. For our present purposes, you may think of a file as a collection of characters. For example, the characters found in Section 9.1 of this book, when stored on diskette, might comprise one file. (In fact, as this book was being written, the sections were stored on diskette in exactly that way.)

A diskette can contain many files. (The maximum number depends on which version of DOS you are using, as well as on the type of drives your computer has.) However, diskette files may be classified into two broad categories—programs and data files.

Programs—A program is a sequence of computer instructions. Throughout this book, we will be discussing programs of one sort or another.

Data files—A data file contains data, such as payroll information, personnel data, recipes, train and airline schedules, appointment calendars, and so forth. Programs often make use of data files. This is done by including instructions within the program for reading (or writing) on particular data files. In this way, you may, for example, look up appointments and let the computer make decisions based on data in the file.

File Names

Each file is identified by a **file name**. Here are some examples of valid file names:

BASIC.COM
 FORMAT.COM
 PAYROLL
 GAME.001

A file name consists of two parts—the main file name (BASIC, FORMAT, PAYROLL, GAME) and an optional extension (COM, COM, no extension, 001). The main file name may contain as many as eight characters, the extension as many as three. The two parts of the file name are separated by a period.

The following characters are allowed in a file name:

The letters A-Z;

The digits 0-9;

Any of these characters:

! @ # \$ % & () - _ [] ' .

Note that a file name CANNOT include any of the following characters:

- \ < > , / ? " ~ : + = * ^

The only period allowed in a file name is the one that separates the two parts of the file name. Moreover, a file name cannot have any spaces.

A file name may be spelled with either upper- or lowercase letters. However, DOS will convert the file name into upper case. So, for example, the file names:

JOHN John

refer to the same file.

TEST YOUR UNDERSTANDING 1 (answers on page 101)

What is wrong with the following file names?

- (a) ALICE 01 (b) #2324/1 (c) alphabetical

A particular diskette can have only one file with a particular name. However, there is nothing wrong with using the same file name to refer to different files on **different** diskettes.

When you name a file, choose a name that somehow suggests the contents. For example, if you generate a monthly payroll file, you can name the various monthly files:

PAYROLL.JAN, PAYROLL.FEB, PAYROLL.MAR, . . .

and so forth.

The Directory

Each diskette has a directory that lists the name of each file on the diskette as well as some descriptive information about the file. Let's analyze one directory entry, the one for the file named DISKCOPY.COM. Actually this file is a program, and one that we've already used, namely the program for copying the contents of one diskette onto another. We used this program to backup our master DOS diskette. The directory entry for DISKCOPY.COM reads:

```
DISKCOPY COM 2444 3-08-83 12:00p
```

The first two parts of the entry, namely DISKCOPY and COM, give the file name and the extension. The next part of the entry gives the size of the file in bytes. DISKCOPY.COM is 2444 bytes long. The final two parts of the directory entry give the date and time the file was last changed. In this example, DISKCOPY.COM was last altered on 3-08-83 at 12:00PM.

DOS maintains the directory automatically. Each time a file is added or changed, DOS makes the appropriate changes in the directory, so that it always accurately reflects the contents of the diskette.

You may examine the directory of a diskette using the DOS command DIR. For example, to examine the directory of the disk currently in the diskette drive, you would type:

```
DIR
```

and press ENTER. The directory of drive A: will then be displayed on the screen.

TEST YOUR UNDERSTANDING 2 (answer on page 102)

Display the directory of your DOS diskette.

Exercises (answers on page 152)

Which of the following file names are valid? If invalid, tell why.

- | | |
|-----------------|----------------------|
| 1. SALLY.001 | 2. EXAMPLE.TXT |
| 3. E> | 4. S:001 |
| 5. #\$\$&{ } | 6. A.B.C |
| 7. ACCOUNT.0123 | 8. DEMONSTRATION.823 |

ANSWERS TO TEST YOUR UNDERSTANDING

- 1: (a) Illegal space
 (b) Illegal character (/)
 (c) Too many characters

2: Place the DOS diskette in the current drive, type:

DIR

and press ENTER.

9.2 File Specifications

As we mentioned in the preceding section, two different files may have the same name as long as they are on separate diskettes. But what happens when you put one of the diskettes in drive A: and one in drive B:. Clearly, there is potential for dangerous confusion here. In order to specify a file without confusion, you must give, in addition to the file name, the location of the file.

PC Device Names

DOS specifies the various components of the PC using the following abbreviations:

A:, B:, C:, D: disk drives (diskette drives or hard disks)

CON: the console (or keyboard)

SCRN: the screen

LPT1: printer #1

LPT2: printer #2

COM1: communications adapter #1

COM2: internal modem

Note that DOS 2.1 allows for four diskette drives even though the PCjr has only one. This is because the same version of DOS 2.1 as you use on the PCjr is also used on the PC and PC/XT. On the PCjr, your single drive is the one named A:. However, to clarify certain commands that would ordinarily involve two drives, your single drive will often be considered as drive B:. For example, if you wish to copy a file from one diskette to another, you can specify a copy operation from drive A: to drive B:. While DOS is reading the file from your source diskette, the drive is called A:. When DOS must write the copy of the file, the diskette drive will become drive B:. You will be directed to insert the target diskette into drive B:. After the copy operation is complete, the identity of the drive once again becomes A:.

Note also that PCjr systems will have only one printer attached. However, DOS allows for expansion.

File Specifications

The combination of a device name and a file name is called a **file specification**. Here are some examples:

```
A:ACCOUNTING
COM1:XYZ.01
LPT1:PRINT.2
```

Each file has an associated file specification that tells the location of the file and the file name. The file specification contains enough information to find the file without any problems.

In order for a file specification to be valid, both the device name and the file name must be valid.

Wild Card Characters

The characters * and ? in a file name have a special meaning for DOS.

The character * may be used as either the main file name or the extension. The portion of the file name replaced by * may be anything at all. For example, consider this file name:

```
*.COM
```

It stands for any file with the extension COM. Similarly, the file name:

```
WS.*
```

stands for any file that has as its main name WS. Finally, the file name:

```
*.*
```

stands for any file.

Such “ambiguous” file names can shorten various DOS commands. For example, we may copy a file from one diskette to another using the COPY command. (See Section 4.) To copy all files from the diskette in drive A: to the diskette in drive B:, we give the simple command:

```
COPY A:*. * B:
```

Similarly, to copy all files on the diskette in drive A: having an extension COM onto the diskette in drive B:, we use the command:

```
COPY A:*.COM B:
```

The character ? in a file name allows a single character to be ambiguous. For example, consider this file name:

```
EXAMPLE.OO?
```

The third letter in the extension may be anything. Similarly, consider this file name:

NGI?? .000

The main file name has five letters, begins with NGI, and the last two letters of the main file name may be anything.

File names using * and ? may be used in file specifications.

Exercises (answers on page 152)

Write file specifications for the following files:

1. ALICE.3 on COM1:
2. MESSAGE on LPT1:
3. Any file on the diskette in drive A:.
4. Any file on the current drive.
5. A file on drive A: in which the main file name has only one letter.
6. A file on drive A: in which the main file name is RALPH and the extension may be anything.

9.3 Executing Commands and Programs

DOS has many commands that perform “housekeeping” functions for the PC. For example, we have already met the command DISKCOPY that lets you copy the contents of one diskette onto another, and the DIR command, which displays a diskette directory. In the rest of this chapter, we’ll discuss the various DOS commands and how to use them. However, let’s first make some general comments about DOS commands.

To execute a DOS command, the DOS prompt must be displayed. Then do the following:

1. Type the DOS command.
2. Press ENTER.

DOS will execute the command. When execution is complete, DOS will redisplay the DOS prompt.

We have already met the above procedure in our discussions of the DOS commands DIR and DISKCOPY.

Before you press ENTER, you may use the BACKSPACE key to correct mistakes. If you make an error in a command, DOS is quite tolerant. For example, give the command XXXXXX. (There is no such command.) DOS will respond with:

```
Bad Command or File Name
A>
```

You may now give another command.

Before you press ENTER, you may press the ESC key. This puts a / on the current line, moves the cursor to the next line and allows you to retype the command.

Here's another type of error that can occur. Remove the DOS diskette and type the command DISKCOPY A: B:. DOS will attempt to read the DISKCOPY program from the non-existent diskette. After a few seconds, DOS will respond with the prompt:

```
Error reading drive A:  
Repeat(R), Ignore(I), or Abort(A)
```

Type R to repeat the command (presumably after you have replaced the diskette), I to ignore the error (in this case, you will generate the same error message), or A (in which case the command will be canceled and the DOS prompt redisplayed). Note that in typing R, I, or A, you don't need to press ENTER.

Internal vs. External DOS Commands

We have already noted that the command DISKCOPY is contained in the file DISKCOPY.COM on the DOS diskette. However, if you inspect the directory of the DOS diskette, you won't find a file named anything like DIR.COM. So where does DOS obtain the program corresponding to the command DIR? The answer is in the way DOS works.

When you start your computer system, you read part of DOS into RAM. This portion of DOS stays in RAM throughout your session with the computer. The most important DOS commands are contained in this portion of DOS so that they can be available without getting them from diskette. Such commands are called **internal commands** and DIR is an example. You may remove the DOS diskette from the current drive and still have the internal DOS commands available.

It would be nice to have all of DOS' commands in RAM all the time. For one thing, they would execute more quickly. However, this gain must be balanced against the permanent decrease in the amount of RAM. Any decrease in RAM would lower the allowable size of application programs. So, as a compromise, the least frequently used DOS commands are stored on diskette. These commands are called **external commands**. When you request an external command, the corresponding diskette file is read into memory and executed. However, upon completion, the memory is made available for the next program or command.

TEST YOUR UNDERSTANDING 1 (answer on page 106)

Remove the DOS diskette from the drive A: . Give the command:

```
DISKCOPY A: B:
```

What happens? Why?

Running Programs Under DOS

We have described the procedure for executing DOS commands. However, the same procedure may be used for programs. Many programs you buy will be stored in files with the extensions COM or EXE. To run such programs, just type the file name without the extension, and press ENTER. For example, the BASIC language is one of the programs on the DOS diskette and its file name is BASIC.COM. So to run BASIC, type BASIC and press ENTER.

ANSWER TO TEST YOUR UNDERSTANDING

1: DOS reports an error reading drive A:, since it cannot find the file DISKCOPY.COM to read.

9.4 The COPY Command

You may move a file from one place to another within the computer using the COPY command. As you might guess, this command is very important. Here is how to use it.

1. Obtain the DOS prompt A>. (If you are in BASIC, type SYSTEM and press ENTER to obtain this prompt.)
2. To copy file specification <filespec1> to file specification <filespec2>, type:

```
COPY <filespec1> <filespec2>
and press ENTER. (Note that there is a space between the two file
specifications.)
```

For example, to copy A:BASIC.COM (this is the copy of the BASIC language BASIC.COM found on the diskette in drive A:) to drive B:, just type:

```
COPY A:BASIC.COM B:BASIC.COM
```

and press ENTER. The computer will make a copy of BASIC.COM on the diskette in drive B:.

Actually, if you want to leave the file name the same in the copy, you may include only the device name in the second file specification. For example, the above copying operation could also be accomplished by typing:

```
COPY A:BASIC.COM B:
```

followed by ENTER.

Creating a Diskette File

We can use the COPY command to copy from the keyboard (device name CON:;) directly to a diskette file. Here's how. Sit down at your computer and obtain the DOS prompt A>. Type:

```
COPY CON: A:TEST
```

and press ENTER.

We have just told DOS that we want to copy a file from the console (keyboard) to drive A: and give the resulting file the name TEST. Now type:

```
This is a test.
```

```
We are creating the file TEST on drive A:.
```

End each line with ENTER. Note that the above lines are displayed on the screen. Moreover, DOS temporarily stores input lines in RAM. To indicate that we are done inputting data, press function key F6, followed by ENTER. DOS will now copy the input lines from RAM to a diskette file, as you requested. You will see the drive light on drive A: go on. This means that the writing operation is in progress. The computer will finally respond with the message:

```
1 File(s) copied
```

You have just created the file TEST. If you are not convinced, list the directory of drive A: by typing:

```
DIR A:
```

followed by ENTER. Among the data appearing on the screen will be a line in this form:

```
TEST 62 2/25/83 11:15a
```

This **directory entry** tells you that the name of the file is TEST, that it contains 62 bytes (62 characters, counting spaces, ENTERs, and so forth.) The file was created on 2/25/83 at 11:15AM. (The computer will compute the date and time from the data you specified when you turned the computer on.)

If you are still not convinced that you have created a file, let's copy the file back to the screen. Type:

```
COPY TEST CON:
```

and press ENTER. We have just requested that DOS copy TEST from the current drive (A:) to the console. (To the computer's way of thinking, the console consists of both the keyboard and screen.) Note that the contents of the file will be displayed on the screen.

Finally, let's copy the file TEST to the printer with the command:

```
COPY TEST LPT1:
```

followed by ENTER. (Before giving the command, check that the printer is on.) The printer will print the contents of the file.

TEST YOUR UNDERSTANDING 1 (answer on page 109)

Create a file TEST2 on drive B: containing the following data:

This line is part of TEST2 on drive B:

TEST YOUR UNDERSTANDING 2 (answer on page 109)

Redisplay TEST2 on the screen.

Using Wild Card Characters with COPY

The wild card characters ? and * are very useful in describing COPY operations. Recall that the character * replaces any sequence of characters within a main file name or an extension. For example, the file name *.001 refers to all files with an extension of 001. Some examples of file names that qualify are:

JANE.001 HOWARD.001 MONEY.001 A.001

A command of the form:

COPY A:*.001 B:

will copy all files on A: with extension 001 onto B:.

To copy all the files on A: to B:, you may use this command:

COPY A:*. * B:

Recall that the wild card character ? stands for a single character. Thus, for example, the file name ??ME.001 can stand for the file names FAME.001 and NAME.001, as well as LAME.001.

TEST YOUR UNDERSTANDING 3 (answer on page 109)

Write a command that copies all files on B: with an extension BAS to A:

Exercises (answers on page 152)

Write DOS commands to:

1. Print B:TEST.
2. Copy all files with the extension COM from A: to B: .
3. Display A:TEST.

4. Copy A:TEST to B: with the new name TEST3.
5. Copy all files from A: to B: whose file names begin with a D and where the main file name has eight characters.

ANSWERS TO TEST YOUR UNDERSTANDING

1. Obtain the DOS prompt A> . Type the line followed by ENTER. Press F6 followed by ENTER.
2. Obtain the DOS prompt. Type COPY B:TEST2 CON:
3. COPY B:*.BAS A:

9.5 COPYing and FORMATing Diskettes

In Chapter 8 we made several copies of the DOS master diskette. However, this diskette, important as it is, is not the only diskette we will need. Indeed, the DOS diskette has very little unused space. We need a diskette with plenty of room to write our own programs and data files. In this section, we'll learn to prepare such diskettes.

Formatting a Diskette

When DOS writes on a diskette, it does so in a very orderly fashion. Data is written in circular rings called **tracks**. (See Figure 9-1.) Each track is divided into a number of sectors (eight or nine depending on the option you choose when creating the tracks). (See Figure 9-2.)

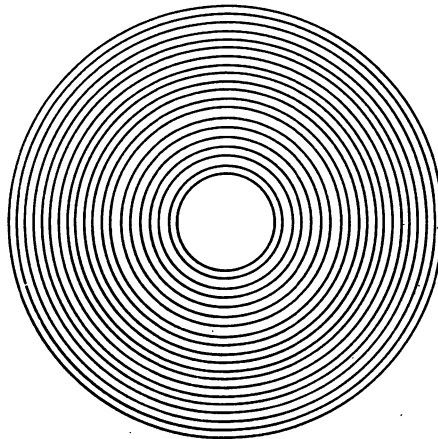


Figure 9-1. The tracks of a diskette.

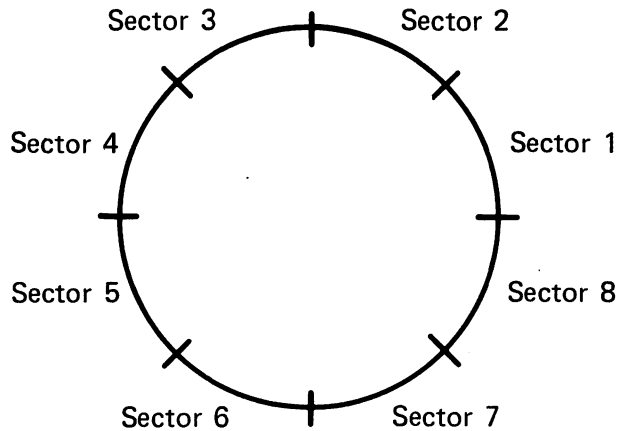


Figure 9-2. The sectors of a diskette track.

In order for DOS to write on a diskette, the track and sector boundaries must be written on the diskette. The IBM PCjr uses **soft-sectored diskettes**, which means that the tracks and sector boundaries are not pre-recorded at the factory. Rather, it is *your* job to prepare a diskette for use by first writing these boundaries on it. This task is called **formatting** and is carried out by the DOS command **FORMAT**.

The **FORMAT** command is an external command. To use it, you must start with a DOS diskette in the current drive. Type:

```
FORMAT <drive>
```

Here <drive> is the name of the drive that will contain the diskette to be formatted. For example, to format a diskette in drive A:, you would type:

```
FORMAT A:
```

DOS will respond with the prompt:

```
Insert new diskette for drive A:
and strike any key when ready
```

Place a blank diskette into drive A:. (If the DOS diskette is in drive A:, don't worry about removing it. The **FORMAT** program is already in RAM at this point.) Press any key. The computer will proceed to format the diskette. Eventually, the display will look something like this (this display is correct for a single-sided drive):

```
Formatting...Format Complete
```

```
362496 bytes on diskette
362496 bytes available
```

```
Format another (Y/N)?
```

At this point, you may answer the question Y(=YES) and format another diskette or N(=NO), in which case DOS will terminate the FORMAT operation and redisplay the DOS prompt, awaiting your next command.

The above procedure was designed so that you can format diskettes one after another. It's a good idea to format an entire box of diskettes when you first buy it. By doing this, you know that all the blank diskettes you have lying around are ready for use.

The numbers displayed in your final FORMAT prompt may be different from those above. The FORMAT command will automatically format your diskette as a two-sided diskette. If, however, you are formatting a diskette on a two-sided drive but wish to then use it on a one-sided drive (say on a PC), you must instruct DOS to format the diskette as single-sided. This is done using the /1 option. For example, to format the diskette in drive A: as a single-sided diskette, use the command:

```
FORMAT A: /1
```

If you are formatting a diskette that might be used with DOS 1.1 (on a PC), then you'll need to format the diskette so that it has only 8 sectors per track. This can be done with the command:

```
FORMAT A: /8
```

You may use the /1 and /8 options together to produce a diskette that is single-sided and 8 sectors per track. In any case, don't be concerned about these alternate diskette formats. They will run on your PCjr without any problems. The only effect will be on the number of bytes of storage available on the formatted diskette.

NOTE You may reformat a diskette that has already been formatted. This erases all data on the diskette. (This is a sure way of destroying sensitive information you don't want lying around.)

TEST YOUR UNDERSTANDING 1

Format a blank diskette.

The number of bytes available will usually be the same as the number of bytes on the diskette. Occasionally, a diskette will contain microscopic flaws that prevent DOS from formatting some sectors. DOS hides these sectors in an invisible file called BADTRACK. You never need to worry about these sectors being used in one of your files and ruining your data! However, if any sectors are placed in BADTRACK, the number of bytes available on the diskette is reduced.

The diskettes produced by the above procedure are totally blank. In particular, they do not contain the DOS files necessary to start the computer. You

may include the DOS internal commands on the formatted diskette by using this command:

```
FORMAT A: /S
```

A diskette produced by this command may be used to start the computer. The DOS internal commands occupy a rather small portion of the diskette. Therefore, most of the diskette is available for your use. When you format a diskette with the /S option, the final display looks something like this:

```
Formatting...Format Complete
```

```
362496 bytes on diskette
 39936 bytes used by system
322560 bytes available
```

```
Format another (Y/N)?
```

TEST YOUR UNDERSTANDING 2 (answer on page 112)

Format a blank diskette with the /S option.

a.

Use this diskette to restart the computer.

b.

Display the directory of A: . Can you explain what you see?

In Chapter 8, we learned to copy a diskette using the DISKCOPY command. We used this command to make a copy of the master DOS diskette. However, we made no mention of FORMATTING in that discussion. The reason is that the DISKCOPY command automatically formats the diskette onto which it is copying (the **target diskette**). This formatting is performed only if necessary.

ANSWER TO TEST YOUR UNDERSTANDING

- 2: (a) Place the formatted diskette in drive A: and press CTRL-ALT-DEL simultaneously.
- (b) The only file in the directory is COMMAND.COM, which occupies 17,664 bytes (in DOS 2.10). The system occupies 39,936 bytes. The remaining bytes are contained in the main DOS files, called IBMBIOS.COM and IBMDOS.COM, which are invisible as far as the directory is concerned.

9.6 Other DOS Internal Commands

This section gives a brief survey of the most commonly used DOS internal commands. Remember that these commands may be used whenever the DOS prompt is displayed. They do not require any information from the DOS diskette.

ERASE—Lets you erase a file. For example, to erase the file EXAMPLE.TXT on the diskette in drive A:, you could use the command:

```
ERASE A:EXAMPLE.TXT
```

If the drive designation is omitted, then the current drive is assumed. For example

```
ERASE EXAMPLE.TXT
```

will erase EXAMPLE.TXT on the diskette in the current drive. The erase command may be used with the wild card characters * and ?. For example, to erase all files on the diskette in drive A:, use the command:

```
ERASE A:*.*
```

TEST YOUR UNDERSTANDING 1 (answer on page 114)

Write a command that erases all files on the current drive with a five-character main name and an extension of COM.

RENAME—Allows you to rename a file. For example, to rename A:OLDFILE with the name NEWFILE, you could use the command:

```
RENAME A:OLDFILE NEWFILE
```

Note that the current file name comes first and then the new file name. If you do not give a drive designation with the current file name, then the current drive is assumed.

TEST YOUR UNDERSTANDING 2 (answer on page 114)

Write a command which renames A:TEST.COM to A:T.COM .

DATE—Lets you set the date. For example, to set the date to 4-12-83, you could use the command:

```
DATE 4-12-83
```

TEST YOUR UNDERSTANDING 3 (answer on page 114)

Write a command which sets the date to Dec. 12, 1984.

TIME—Lets you set the time. For example, to set the time to 1:04:00 PM, you could use the command:

```
TIME 13:04:00
```

TEST YOUR UNDERSTANDING 4 (answer on page 114)

Write a command to set the time to 12:00:00 AM.

TYPE—Allows you to display the contents of a file. For example, to display the contents of the file A:TEST1, you could use the command:

```
TYPE A:TEST1
```

If you try to display a program, it will usually look like a bunch of gibberish. Program files are designed for the convenience of the computer, not for humans. However, a text file will be displayed in readable form.

To obtain a written copy of a file, first press Ctrl-PrtSc. Then give the type command. The file will be displayed on the screen and also printed on your printer.

COMP—Compares two files to determine whether they are identical. For example, suppose that we wish to compare FILE1 on the diskette in drive A: with FILE2 on the diskette in drive B:. Give the command:

```
COMP A:FILE1 B:FILE2
```

This command can be used to check on the results of a COPY operation to determine whether the copy is identical to the original. Note that the COMP command does not give you a chance to change diskettes. Therefore, the diskettes with the files needed for comparison must be in the diskette drive prior to giving the COMP command.

ANSWERS TO TEST YOUR UNDERSTANDING

- 1: ERASE ??????.COM
- 2: RENAME A:TEST.COM A:T.COM
- 3: DATE 12-12-84
- 4: TIME 00:00:00

9.7 Other DOS External Commands

In this section, we summarize some of the most commonly used DOS external commands. Note that, in order to use any of these commands, DOS must obtain the appropriate program from the DOS diskette.

DISKCOMP—Allows you to compare the contents of two diskettes, byte by byte. For example, to compare the diskettes in drives A: and B:, you could use the command:

DISKCOMP A: B:

If your system has only one drive, you would also use this command for diskette comparison, even though you don't have a drive B:. DOS will prompt you to swap the diskettes in your single drive so that a comparison can be made.

CHKDSK—Allows you to check on the number of bytes remaining on a diskette. It also performs a check to determine if any inconsistencies exist in the way the files are stored. To perform a CHKDSK operation on the diskette in drive A:, you could use the command:

CHKDSK A:

The result of this command is a display of the form:

```
362496 bytes total disk space
22272 bytes in 2 hidden files
45455 in 4 user files
294769 bytes available on disk
```

```
131072 bytes total memory
118321 bytes free
```

As usual, your numbers may vary, depending on your system, how your diskette has been formatted, and so forth.

You should execute a CHKDSK every so often for each of your diskettes, to assure the integrity of your files and to determine the space remaining on the diskette.

TEST YOUR UNDERSTANDING 1 (answer on page 115)

Suppose that the DOS diskette is in drive B: and that drive A: is the current drive. Write a command for performing CHKDSK on the diskette in drive A: .

ANSWER TO TEST YOUR UNDERSTANDING

1: B:CHKDSK A:

9.8 Creating Your Own DOS Commands—Batch Files

In the preceding sections, we learned about the most useful DOS commands. Most often, you will execute DOS commands by typing them directly from the keyboard, as described earlier in the chapter. However, in many applica-

tions, it is necessary to execute the same sequence of DOS commands repeatedly. For example, consider the following situation.

Suppose that you have a diskette containing four files named `ACCOUNTS.MAY`, `PROFIT.MAY`, `PAYABLE.MAY`, and `SALES.MAY`. Your business is computerized and every one of your 10 managers has an IBM PC. Rather than distribute the contents of the files via paper copies, in the traditional manner, you wish to send each manager a copy of the files on diskette.

A simple solution would be to use `DISKCOPY` to make 10 copies of the diskette containing the files. Suppose, however, that your diskette also contains some sensitive information that you do not wish to circulate. In this case, you may prepare the duplicate diskettes by copying the files one at a time. This may be done via the `COPY` command. Here are the DOS commands required to prepare one duplicate diskette, starting from an unformatted diskette.

```
FORMAT B: /S
COPY A:ACCOUNTS.MAY B:
COPY A:PROFIT.MAY B:
COPY A:PAYABLE.MAY B:
COPY A:SALES.MAY B:
```

Assume that your files are contained on the same diskette as `FORMAT.COM` and that this diskette is in drive A:. The duplicate diskette is in drive B:.

It is possible to prepare the 10 duplicates by typing these commands in manually. But what a chore! And it is easy to make a mistake in typing, especially as the afternoon wears on. There is, fortunately, a much better way to proceed, namely to use a batch file.

A **batch file** is a diskette file consisting of a list of DOS commands. A batch file must have a file name with the extension `BAT`. In our case, let's name the batch file `C.BAT`, and let's store it on the diskette in drive A:. In order to create the batch file, use the `COPY` command. Type:

```
COPY CON: A:C.BAT
```

and press `ENTER`. Now type in the DOS commands, exactly as they appear in the above list. At the end of each line, press `ENTER`. After the last line and its `ENTER`, press function key `F6` and then `ENTER`. DOS will then respond with the message:

```
1 file(s) copied
```

The file `C.BAT` is now on the diskette in drive A:.

To execute the list of DOS commands, we now merely type the letter `C` and press `ENTER`. (It is just as if we created a new DOS command with the name

C.) DOS will then search the current diskette (A:), find the batch file and execute the various commands, in the order specified.

TEST YOUR UNDERSTANDING 1 (answer on page 118)

Modify the above list of DOS commands so that they include a check that the copies of the files are identical to the originals.

Now our copying job is cut down to size:

1. Insert a blank diskette into drive B:.
2. Type C and press ENTER.
3. Wait for the commands to be executed.
4. Repeat operations 1-3 until all 10 copies are made.

The AUTOEXEC.BAT File

The AUTOEXEC.BAT file is a batch file that is automatically executed whenever DOS is started. If a diskette contains a file with the name AUTOEXEC.BAT, then it is executed on DOS startup without any operator action. For example, suppose that you want your PCjr to start BASIC automatically whenever DOS is started. Just create a diskette file called AUTOEXEC.BAT containing the command:

BASIC

Note that you may have only one AUTOEXEC.BAT file on a given diskette. On the other hand, you may have many batch files.

TEST YOUR UNDERSTANDING 2 (answer on page 119)

Modify your DOS diskette so that BASIC is started whenever you start DOS.

The AUTOEXEC.BAT file may be used for some clever purposes. For example, let's return to our company with 10 managers. Suppose that you wish to include a covering memo that reads:

```
TO:MANAGERS
HERE ARE THE STATEMENTS FOR MAY.
WE'LL MEET TO DISCUSS THEM ON 6/4
AT 5:30 pm.
JR
```

Here is how the message can be displayed automatically:

1. Create a file on your diskette that contains the text of the message. Call the file MSSG.

2. Create a file AUTOEXEC.BAT that contains the DOS command:

```
TYPE MSSG
```

3. Modify the batch file C.BAT so that it copies MSSG and AUTOEXEC.BAT onto each of the 10 copies.

Each manager will start his or her PC using a duplicate diskette. The AUTOEXEC.BAT file will cause the file MSSG to be displayed on the screen.

Parameters

Let's stick with our fictitious company. Suppose that the 10 diskettes are to be prepared and sent every month. The file names are always the same, but the month abbreviations, as given in the file name extensions, vary. You could prepare a new batch file C.BAT every month. However, there is a better way. Designate the month abbreviation by the symbol %1. (% is an abbreviation for **parameter** and 1 is the number of the parameter.) The commands of the batch file are then written:

```
FORMAT B:/S
COPY A:ACCOUNTS.%1 B:
COPY A:PROFIT.%1 B:
COPY A:PAYABLE.%1 B:
COPY A:SALES.%1 B:
```

For the month of MAY, we would give the batch command:

```
C MAY
```

For the month of JUNE, give the batch command:

```
C JUN
```

And so forth.

You may use up to nine parameters (%1, %2, ..., %9). You specify the values of these parameters when you give the batch command, with consecutive parameter values separated by spaces. For example, if a batch file D used the two parameters %1 and %2, then to execute the batch file with %1 = JAN and %2 = FEB, we would use the command:

```
D JAN FEB
```

ANSWERS TO TEST YOUR UNDERSTANDING

- 1: Add the DOS commands:

```
COMP A:ACCOUNTS.MAY B:
COMP A:PROFIT.MAY B:
```

COMP A:PAYABLE.MAY B:

COMP A:SALES.MAY B:

- 2: Add the file AUTOEXEC.BAT which consists of the single DOS command:

BASIC

10

MORE ABOUT DOS 2.10

10.1 Paths, Directories and Subdirectories

Directories and Subdirectories

DOS 2.10 allows you to manipulate files contained on a hard disk drive, such as the one contained in the system unit of the PC/XT. As far as the user is concerned, a hard disk functions much like a floppy disk, except for its greater storage capacity and its higher speed. However, a hard disk may contain thousands of files at one time. A single directory would be unwieldy. It would be difficult for you to look at and it would be slow for the computer to search for a particular file. To solve this problem, DOS 2.10 allows you to organize your directory in a "tree structure." If you wish, you may ignore this complication on the PCjr. However, the tree structured directories of DOS 2.1 are useful even on diskette, where they allow you to organize your directories in a logical fashion.

To explain what we mean, consider the typical DOS 2.10 directory shown in Figure 10-1.

```
Volume in drive C is DOS
Directory of C:\MIDWEST

.                <DIR>          4-17-83   7:49p
..               <DIR>          4-17-83   7:49p
PAYROLL      83   <DIR>          4-30-83  10:09a
PERSON       83   <DIR>          5-12-83  11:00a
INVENTORY   83   <DIR>          5-12-83   3:45p
MEMO        518   5847          5-18-83   1:12p
MEMO        520   4850          5-20-83   3:24p
```

Figure 10-1. A DOS 2.10 directory.

The first line of the directory gives the volume title. DOS 2.10 allows you to assign a title to a disk. In this case, the contents of the hard disk are named

“DOS.” (Drive C: is the hard disk.) A typical DOS 2.10 directory has two sorts of entries, files and directories. A file entry is exactly the same sort of entry as used in DOS 1.1. For example, the directory of Figure 10-1 lists the files MEMO.518 and MEMO.520. The notations <DIR> indicate that all of the other entries are directories: PAYROLL.83, PERSON.83, INVENTORY.83. Each of these directories are **subdirectories** of the directory shown. They may each contain files and directories of their own. (Don’t worry about the directories “.” and “..” for now. They serve a purely technical role.)

In DOS 2.10, each directory has a name, which may consist of a main part and an extension. The rules for directory names are the same as for file names. (The main name may contain up to eight characters, the extension up to three, and so forth.) For example, the directory shown in Figure 10-1 has the name MIDWEST.

The organization of files into directories may be described as a **tree**. For example, Figure 10-2 shows the relationship of the directory of Figure 10-1 with other directories on the same disk. The main directory RECORDS has three subdirectories: MIDWEST, SOUTH, and NORTHEAST. Each of these subdirectories has several subdirectories as well as several files.

Note that a file or directory name may be repeated in various parts of a tree. For example, each of the directories MIDWEST, SOUTH, and NORTHEAST has a subdirectory named PERSON. However, each directory named PERSON is independent; their respective contents need have nothing to do with one other.

The Root Directory

When a diskette is formatted, DOS creates a master directory called the **root directory**, designated by the symbol \ . (This symbol is called a **backslash**.) All directories are subdirectories of the root directory. Note the root directory at the bottom of the tree in Figure 10-2.

The Current Directory

At any given moment, each disk drive has a directory designated the **current directory**. If you give a DOS command without specifying a directory, then the current directory is assumed. When DOS is started, the current directory is the root directory for each drive.

Here are some examples of DOS commands that refer to the current directory:

```
C:  
DIR A:
```

The first command changes the current drive to C: and the directory to the current directory of C:. The second command lists the contents of the current directory of the diskette in drive A:.

Free Subscription!

Welcome to PCjr PARTYLINE™ —a brand new information service for IBM PCjr users.

What you'll get is a personal report, updated quarterly, covering the latest PCjr books, games and software available from Brady. Current, informative and easy-to-use, PCjr PARTYLINE™ offers you valuable *advance* information on forthcoming new products.

To begin your *FREE* subscription, complete and mail this card today! No postage (or purchase) required.

(Please print or type)

Name _____

Address _____

City _____ State _____ Zip _____

How do you usually find out about new books and software?

- Magazine Advertisements Mail at Work or Home
 Book/Computer Store Browsing Friend's Recommendation
 Other (please explain) _____

What areas are you most interested in?

- Business Applications Education
 Home Applications Games
 Other (please explain) _____

Today's Date _____ Signature _____

BRADY BOOKWARE™

Robert J. Brady Co. • Bowie, MD 20715

A Prentice-Hall Company

D5467-8



**NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES**

BUSINESS REPLY CARD

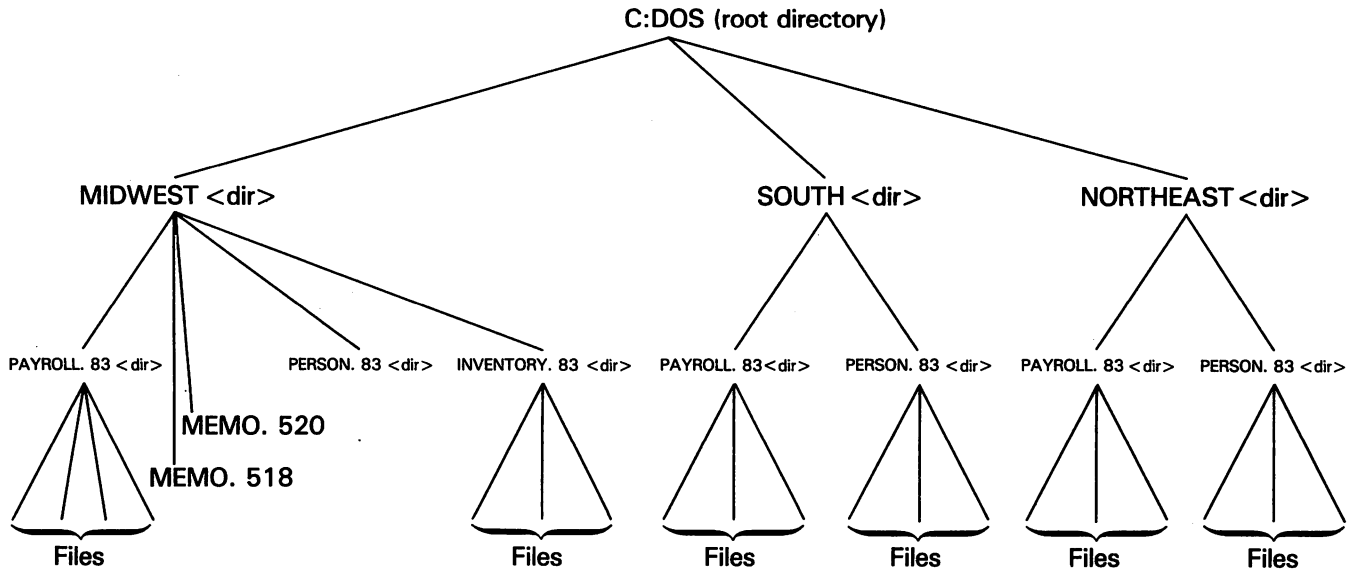
FIRST CLASS PERMIT NO. 1976 BOWIE, MD.

Postage will be paid by addressee

**Robert J. Brady Co.
A Prentice-Hall Company
Bowie, Maryland 20715**



Figure 10-2. The directory of C:.



Paths

To completely specify a file for use in a DOS command, you must give the drive, the path, and the file name to the file starting from the current directory. For example, suppose that the current directory is MIDWEST and you wish to refer to FILE.1. You would specify the file with:

```
INVENTORY\FILE.1
```

This sequence of names is called a **path**. It tells DOS that to find the file, start from the current directory, go to the subdirectory INVENTORY, and then to FILE.1. As a second example, suppose that the current directory is the root directory. Then the path to FILE.1 is:

```
MIDWEST\INVENTORY\FILE.1
```

As a third example, suppose that the current drive is A:. The path to FILE.1 is now:

```
C:\MIDWEST\INVENTORY\FILE.1
```

(The path begins with \ to indicate that the search should begin with the root directory.)

Most DOS commands may use paths instead of file names. For example, suppose that the current directory is MIDWEST. Then the command:

```
DIR INVENTORY
```

will list the contents of the subdirectory INVENTORY. Similarly, suppose that the current drive is A:. Then the command:

```
COPY C:\MIDWEST\INVENTORY\*.* A:
```

will copy all files in the subdirectory C:\MIDWEST\INVENTORY to the current directory of the diskette in drive A:. Any missing parameters in a path mean that the current value of the parameter is to be assumed. For example, if the drive designation is omitted, then the current drive is assumed. If the path is omitted, then the current directory is assumed.

TEST YOUR UNDERSTANDING 1 (answer on page 128)

- (a) Write a command to list the directory of NORTH\PERSON.
- (b) Write a command to copy all files in NORTH\PERSON to SOUTH\INVENT.

TEST YOUR UNDERSTANDING 2 (answer on page 128)

What does this command do?

```
COPY A: *.* C:
```

TEST YOUR UNDERSTANDING 3 (answer on page 128)

What does this command do?

COPY *.* A:

The symbol .. may be used in a path to indicate a move one level down the tree, from a subdirectory to its “parent directory.” For example, suppose that the current directory is MIDWEST. Then the path:

..

refers to the root directory. Similarly, if the current directory is the subdirectory INVENTORY of MIDWEST, then the path:

..\..

describes a move two steps down the tree, to the root directory. As yet another example, suppose that we want to get to the subdirectory PERSON of SOUTH from the subdirectory INVENTORY of MIDWEST. We must go down to the root directory and back up to PERSON. Here is a path that describes these moves.

..\..\SOUTH\PERSON

DOS 2.10 limits the length of paths to 63 characters.

Changing the Current Directory

You may change the current directory by using the DOS command CHDIR. For example, suppose that the current directory is MIDWEST and we want to change to the subdirectory PERSON. The proper command is:

CHDIR PERSON

The general format of the CHDIR command is:

CHDIR <path>

Here <path> is a path from the current directory to the desired current directory.

If a drive designation is included in the path, then CHDIR will change the current directory on the indicated drive. However, the current drive is not changed. For example, suppose that you wish to perform the directory change “..” for drive A: and the current drive is C:. The command:

CHDIR A:..

will change the current directory on A: but will leave the current drive as C:.

Directory changes occur so often that DOS 2.10 allows you to use the shorthand CD instead of CHDIR.

TEST YOUR UNDERSTANDING 4 (answer on page 128)

- (a) Suppose that the current directory is MIDWEST. Write a command that changes the current directory to the root directory.
- (b) Suppose that the current directory is MIDWEST. Write a command that changes the current directory to the subdirectory PERSON of SOUTH.
- (c) Suppose that the current directory is MIDWEST and the current drive is C:. Change the current drive to A: and perform the directory change .. on C: .

Creating and Deleting Directories

You may create directories using the MKDIR command. For example, suppose that the current directory is the root directory. To create the subdirectory MIDWEST, we would use the command:

```
MKDIR \MIDWEST
```

Still assuming that the current directory is the root directory, we may create the subdirectory INVENTORY of MIDWEST using the command:

```
MKDIR \MIDWEST\INVENTORY
```

The general form of the MKDIR command is:

```
MKDIR [drive:]path
```

If the drive is omitted, then the current drive is assumed.

DOS allows you to use the shorthand MD as a substitute for MKDIR.

TEST YOUR UNDERSTANDING 5 (answer on page 128)

Suppose that the current directory is MIDWEST. Write a command to create the directory SOUTH in the root directory.

Note that you cannot copy files into a directory until the directory is created by the MKDIR command. BASIC and DOS will respond to any attempt to reference a non-existent directory with an **Invalid Directory or Filename** error.

The MKDIR command creates the specified directory and enters in it two files, named . and .. . The first of these contains the names of all files and subdirectories contained within the particular directory. Initially, the file . contains no entries. The file .. specifies the directory to which this subdirectory belongs. That is, the file .. contains the information necessary to descend one level in the tree.

The root directory may contain as many as 64 entries in case of a single-sided diskette or 112 entries in case of a double-sided diskette or a hard disk. However, there are no limitations on the number of entries in a subdirectory.

You may delete a directory using the RMDIR command. To remove a directory, it must contain only the files . and .. . (This is a safety feature so that you will not accidentally delete files by mistake.) The RMDIR command works similarly to the command MKDIR. It has the format:

```
RMDIR [drive:] path
```

For example, if the current directory is the root directory, then you may remove the subdirectory PERSON of MIDWEST using the command:

```
RMDIR \MIDWEST\PERSON
```

If you try to remove a directory containing files other than . and .. , then DOS will report an error.

DOS allows the shorthand RM as a substitute for RMDIR.

You can add files to a directory in the usual ways, using the COPY command, for instance. Similarly, you may erase files using the ERASE command. Moreover, COPY and ERASE may use paths to describe the files involved. Of course, you may omit the paths if you want to refer to files in the current directory.

For example, to COPY FILE1 to the current directory of drive C:, we use the command:

```
COPY FILE1 C:
```

Suppose that the current directory is the root directory and we wish to copy FILE1 into the subdirectory MIDWEST\PERSON. We use the command:

```
COPY FILE1 MIDWEST\PERSON
```

After performing this command, we could then erase FILE1 using the command:

```
ERASE FILE1
```

NOTE—The COPY and ERASE commands refer only to files and not to directories. To copy the contents of a directory, you must first use MKDIR to create a directory with the proper name. Then use the COPY command to copy the desired files into the new directory.

TEST YOUR UNDERSTANDING 6 (answer on page 128)

- (a) Write a command to copy all files in the current directory on A: to the directory MIDWEST\INVENTORY .
- (b) Write a command to erase all files in the root directory of drive A:.

ANSWERS TO TEST YOUR UNDERSTANDING

- 1: (a) DIR NORTH\PERSON
(b) COPY NORTH\PERSON*.* SOUTH\INVENT
- 2: It copies all files on the diskette in drive A: onto the current directory.
- 3: It copies all files in the current directory onto the diskette in drive A:.
- 4: (a) CD .. <ENTER>
(b) CD ..\SOUTH\PERSON <ENTER>
(c) A:<ENTER>
CD C:.. <ENTER>
- 5: MD ..\SOUTH
- 6: (a) COPY A:*.* C:MIDWEST\INVENTORY
(b) ERASE A:*.*

10.2 More DOS Commands

Let's look at more DOS 2.1 commands. In describing the format of the various commands, we will enclose in brackets [] input data that is optional.

DATE—Sets the date or to display the current date. It has the format:

DATE [mm-dd-yy]

Here mm-dd-yy is the date in the form month (1-12), day (1-31), and year (80-99). If the date is included, DATE sets the date. Otherwise, the date is displayed.

TIME—Sets or displays the time. It has the format:

TIME [hh-mm-ss]

Here hh-mm-ss is the time in hours (0-23), minutes (0-59), and seconds (0-59). If the time is included, then TIME sets the time. Otherwise, the time is displayed.

SYS—Writes DOS onto a diskette. Space must have been reserved on the diskette, since DOS can reside only in certain sectors. To record DOS on the diskette in drive d:, use the command:

SYS [d:]

If the [d:] is omitted, then DOS will be written onto the diskette in the current drive.

The commands **COMP**, **DIR**, **DISCOMP**, **DISKCOPY**, **RENAME**, and **TYPE** allow the use of paths in addition to file names. Let's describe this further capability of these commands.

COMP—Compares two files to determine whether they are the same. Here is the format of the **COMP** command.

```
COMP [d:][path][filename] [d:][path][filename]
```

If you omit either of the two sets of file data, DOS will prompt you to supply them. In DOS 2.10, you may use wild card characters * and ? in the file data to compare several files in one operation.

DIR—Lists the directory corresponding to a given set of file data. Here is the format of the **DIR** command.

```
DIR [d:][path][filename] [/P] [/W]
```

You may use this command to display the directory information about a single file or about a collection of files described by a path. The /P option tells DOS to pause after each screenful of data. You may continue with the listing of file data by pressing any key. The /W option displays the files by name only and in a wide format, allowing you to view the contents of especially large directories.

DISKCOMP—Compares the contents of two diskettes. Here is the format of the **DISKCOMP** command.

```
DISKCOMP [d:] [d:] [/1] [/8]
```

If you omit a drive designation, then the current drive is assumed. The /1 option allows you to compare only the first sides of two-sided diskettes. The /8 option allows you to compare only 8 sectors per track (sectors 1-8) even if the diskettes have been formatted with 9 sectors per track.

DISKCOPY—Copies the contents of one diskette onto another. Here is the format of the **DISKCOMP** command.

```
DISKCOPY [d:] [d:] [/1]
```

If you omit a drive designation, then the current drive is assumed. The /1 option allows you to copy only one side of a two-sided diskette. If the target diskette is not formatted, then the **DISKCOPY** command will also format. It will format the target diskette in exactly the same format as the source diskette.

RENAME—Renames a file. Here is the format of the **RENAME** command.

```
RENAME [d:][path][filename] [filename]
```

You may use wild card characters * and ? in the filenames. Any portions of a file name that are unambiguous are carried over from the statement of the

command. The ambiguous characters are determined from the name of the source file.

TYPE—Displays the contents of a file. Here is the format of the TYPE command.

```
TYPE [d:][path][filename]
```

CHKDSK—Checks the directories and file allocation table for inconsistencies and displays a report on unused disk and memory space. Here is the format of the CHKDSK command.

```
CHKDSK [d:][filename][/F][/V]
```

If you specify a filename, CHKDSK will analyze the memory allocated for the designated file. CHKDSK will look only at the current directory, so no path is allowed. The /F option tells CHKDSK to fix any errors that it finds. The /V option tells CHKDSK to report on its progress with the errors it finds and their locations.

The FORMAT command has several options that we haven't yet discussed.

FORMAT—Formats a diskette. Here is the format of the FORMAT command.

```
FORMAT [d:][/S][/1][/8][/V][/B]
```

The /S option allows you to copy the system files onto the diskette. The /1 option allows you to format the diskette as single-sided. The /8 option allows you to format the diskette with 8 sectors per track. The /V option will prompt you for a volume name. The /B option reserves space for any version of DOS. With this option, you may later write DOS 1.00, 1.05, 1.10 or 2.10 onto the diskette with the SYS command. Without the /B option you can place only DOS 2.10 onto the diskette.

The default FORMAT makes the diskette double-sided with 9 sectors per track.

As part of the FORMAT operation, the sectors are verified and any defective sectors are marked so that they won't be included in a file. FORMAT reports on the total number of bytes on the diskette, the number of defective bytes, the number of bytes allocated to system files, and the number of free bytes available.

MODE—Sets parameters for the screen, printer, or asynchronous communications adapter.

Option 1: Printer—Sets parameters for the printer. The format of the MODE command for Option 1 is:

```
MODE LPT#[n][,][m][,][P]
```

where # is the number of the printer (1-3), n is the number of characters per line (80 or 132), m is the number of lines per inch vertically (6 or 8), and P specifies a continuous retry in case of time-out errors.

Option 2: Screen—Sets parameters for the screen. The format of the MODE command for Option 2 is:

```
MODE [n][,m][,T]
```

Here n controls the adapter, color, and line width. The possibilities for n are:

```
40      -color/graphics adapter, 40 characters per line
80      -color/graphics adapter, 80 characters per line
BW40    -color/graphics adapter, no color, 40 characters per line
BW80    -color/graphics adapter, no color, 80 characters per line
CO40    -color/graphics adapter, color, 40 characters per line
CO80    -color/graphics adapter, color, 80 characters per line
MONO    -monochrome display adapter
```

The parameter m lets you shift the display one character right or left; m is either R (right shift) or L (left shift).

The parameter T tells DOS to display a test pattern to align the display.

Option 3: Asynchronous Communications Adapter—Defines the parameters for the communications adapter. The format of the MODE command for Option 3 is:

```
MODE COMn:baud[,parity[,databits[,stopbits[,P]]]]
```

Here:

n = the number of the communications adapter (1-3)

baud = the desired baud rate

parity = N (no parity), O (odd parity), E (even)

databits = 7 or 8

stopbits = 1 or 2

P = connection to a serial printer. Device time-outs are continuously retried.

The current settings are even parity, 7 databits, and 2 stopbits if baud = 110 and 1 stopbit otherwise. Note that the baud must always be specified.

Option 4: Redirection of Parallel Printer To Asynchronous Communications Adapter—For this option, the MODE command has the format:

```
MODE LPT#:=COMn
```

Here # is the number of the printer and n is the number of the asynchronous communications adapter. To use this option, you must first initialize the asynchronous communications adapter by using the MODE command in Option 3.

ASSIGN

This command allows you to redirect all instructions involving one device to another device. For example, the command:

```
ASSIGN A=LPT1
```

reroutes all requests involving drive A to the printer. The command:

```
ASSIGN
```

returns the drive routing to its current setting: A to A, B to B, and so forth. An ASSIGN command remains in effect until another ASSIGN command is given or DOS is restarted. Drive reassignments are ignored by DISKCOPY and DISKCOMP.

TEST YOUR UNDERSTANDING 1 (answer on page 137)

Write a command that assigns all diskette commands for drive C: to drive A:.

The next command is of use only in dealing with a hard disk and is not used on the PCjr. However, it is included for the sake of completeness.

BACKUP

A hard disk may contain literally thousands of files. To guard against possible data loss, it is advisable to keep backup copies of all hard disk data on diskette. The BACKUP command allows you to make such copies with a minimum of operator input.

To use the BACKUP command, you must have formatted diskettes ready with enough room to contain the files to be backed up. The format of the BACKUP command is:

```
BACKUP [d:][path][filename] d:[/S][/M][/A][D:mm-dd-yy]
```

This command instructs DOS to back up the files specified by [d:][path][filename] onto the diskettes to be inserted in drive d: . DOS will instruct you to insert the diskettes, as needed.

The BACKUP process may be quite time-consuming. For this reason, there are several options that let you be selective in the files you back up.

The /S option tells DOS to back up all files in the subdirectories of the indicated path. For example, suppose that the path was:

```
C:MIDWEST
```

where MIDWEST is a subdirectory of the root directory of C:. The /S option tells DOS to back up all files in MIDWEST, all files in any subdirectories of MIDWEST, all subdirectories of those subdirectories, and so forth.

The /M option tells DOS to back up only those files that have been modified since the most recent backup operation. This option is especially useful if you have modified only a few files of a directory since the last backup. Using the /M option may save a lot of time.

The /D:mm-dd-yy operation means you can backup only those files that have been either created or modified since the given date mm-dd-yy.

The /A option tells DOS to add the backup files to the diskette already in the specified drive.

TEST YOUR UNDERSTANDING 2 (answer on page 137)

Write commands to do the following backup operations.

- (a) Back up all files in the current directory of C: .
- (b) Back up all files in the directory C:MIDWEST that have been modified since the last backup.
- (c) Back up the file specified by MIDWEST\INVENT\STATUS.051.
- (d) Back up the files in the directory MIDWEST that have been modified since Dec. 4, 1983.

BREAK

Under default conditions, DOS checks for the Ctrl-Break combination only during screen, keyboard, printer, or communications adapter operations. However, you may instruct DOS to check for Ctrl-Break during any program-requested DOS operations. This would enable you to interrupt a diskette read or write operation, for example. To enable checking for Ctrl-Break, use the command:

BREAK ON

To disable checking, use the command:

BREAK OFF

CLS

The CLS command clears the screen from DOS.

GRAPHICS

The GRAPHICS command allows you to print the screen on the IBM Graphics Printer. This command is invoked by pressing Shift-PrtSc. If the computer is in one of the graphics modes, printing may take up to three minutes. In high-resolution graphics mode, the image is printed sideways on the paper.

PATH

Under default conditions, you may run only programs that are in the current directory, and those programs may make reference only to files in the current directory. However, the PATH command allows you to alter these conditions by specifying alternate paths to search for programs or data files. The format of the path command is:

```
PATH [d:]path;[[d:]path;...]
```

When you issue a command or make reference to a file, DOS will first look in the current directory. If the program or file is not found, DOS will then look in each of the paths specified by PATH, in the order in which the paths are listed.

TEST YOUR UNDERSTANDING 3 (answer on page 137)

Write a command that tells DOS to refer to the directories MIDWEST, NORTH, and A:CURRENT .

PRINT

The PRINT command is a sophisticated file transfer utility that allows you to transfer a copy of a file to any of the devices LPT1:, LPT2:, LPT3:, COM1:, COM2:, AUX:, or PRN:. Moreover, it allows you to specify up to 10 files to be transferred. The transfer may take place while the computer is being used for other tasks. The format of the PRINT command is:

```
PRINT [[d:]filename][/T][C][P]...
```

For example, to PRINT the files FILE1 and FILE2, you would use the command:

```
PRINT FILE1 FILE2
```

The files to be printed must be contained in the current directory. However, after the PRINT command has been given, you may change the current directory without affecting the PRINT operation.

The first time you execute a PRINT command after system startup, DOS gives you an opportunity to identify the receiving device. DOS supplies the prompt:

```
Name of list device [PRN]:
```

You may type any of the device names given above. If you press ENTER, then you are choosing the default device PRN: .

The /T option terminates printing of the current file and cancels all files waiting to be printed.

The /C option selectively cancels printing of the files listed. Cancellation begins with the file name preceding the /C and continues until a /P is encountered. If no /P is encountered, then cancellation applies to all files listed.

The /P option allows you to add files to the print list. A file name followed by /P is added to the list, as are all subsequent files. Adding files continues until a file name is followed by /C.

TEST YOUR UNDERSTANDING 4 (answer on page 137)

Write DOS commands to perform the following operations:

- (a) Print the files FILE1 and FILE2.
- (b) Add the files FILE3 and FILE4 to the print list.
- (c) Cancel printing of FILE4.
- (d) Terminate all printing.
- (e) Print the files to the communications adapter COM1: .

RECOVER

After you write a file on a diskette, the diskette, for one of many reasons, may develop a bad sector that destroys some of the data of the file. The RECOVER command lets you recover the undamaged data from such a file. The format of the RECOVER command is:

```
RECOVER [d:][path][filename]
```

or

```
RECOVER d:
```

In the first format, the RECOVER command reconstructs the data file, inserting blanks for the damaged data. The second format is used to RECOVER all files on a diskette in case the directory is damaged.

The next command is of use only in dealing with a hard disk and is not used on the PCjr. However, it is included for the sake of completeness.

RESTORE

The RESTORE command is the reverse of BACKUP. With it, you can write from a backup diskette to a hard disk. The format of the RESTORE command is:

```
RESTORE d: [d:][path][filename][/S][/P]
```

For example, to restore the contents of the backup diskette in drive A: to the current directory of the hard disk, we could use the command:

```
RESTORE A: C:
```

To restore all files on drive A: with names MEMO.* to the directory MIDWEST, we could use the command:

```
RESTORE A: MIDWEST\MEMO.*
```

The /S option is similar to the /S option of the BACKUP command. It allows you to restore all files in all subdirectories of the indicated directory, as well as all files in sub-sub directories, and so forth.

The /P option tells DOS to prompt you with the names of files that have changed since they were last backed up or that are read-only. You may then decide whether to restore the file.

Note that a BACKUP copy of a file should not be used by a program. The DOS BACKUP command uses all the space on a diskette. If there is not enough space for an entire file, then DOS splits the file between two diskettes. Your program has no way of telling whether or not you have the entire file on the current diskette. So before using a BACKUP copy of a file, do a RESTORE first.

TREE

The TREE command allows you to display the directory of a disk in tree form. The command:

```
TREE d:
```

displays all directories on drive d:. Along with each directory, DOS lists all subdirectories; and with each subdirectory, its subdirectories, and so forth.

The option /F allows you to include all files in the tree listing. For example, to list all directories and files on the disk C:, use the command:

```
TREE C: /F
```

In most cases, the output of TREE will take up more than a single screen. To inspect the output, press Ctrl-Prtsc prior to giving the TREE command. The output will then be sent to both the screen and printer.

VER

This command causes DOS to display the version number in the form:

```
IBM Personal Computer DOS Version 2.10
```

VERIFY

Under default conditions, DOS does not verify the accuracy of disk write and read operations. You may enable or disable disk operation verification using the VERIFY command. To enable verification, use the command:

```
VERIFY ON
```

To disable verification, use the command:

```
VERIFY OFF
```

If verification is enabled, then diskette read and write operations will be slowed considerably.

VOL

The VOL command displays the volume name of a specified diskette. For example, to display the volume name of disk C:, use the command:

```
VOL C:
```

The name is displayed in the form:

```
Volume in drive C: is RECORDS
```

The volume name is the one given when the disk was formatted.

```
VOL [d:]
```

ANSWERS TO TEST YOUR UNDERSTANDING

- 1: ASSIGN C=A
- 2: (a) BACKUP C: A:
(b) BACKUP C:MIDWEST*.* A: /M
(c) BACKUP C:MIDWEST\INVENT\STATUS.051 A:
(d) BACKUP C:MIDWEST A: /D12-04-83
- 3: PATH MIDWEST;NORTH;A:CURRENT
- 4: (a) PRINT FILE1 FILE2
(b) PRINT FILE3/A FILE4
(c) PRINT FILE4/C
(d) PRINT /T
(e) Answer the initial prompt with COM1: and press ENTER.

10.3 Enhanced Batch File Capability

DOS 2.10 has batch file capabilities that we haven't yet explored. In this section, we'll survey these capabilities.

DOS 2.10 batch files control the execution of sequences of DOS commands and/or user programs. The batch files of DOS 2.10 are actually programs in their own right. In DOS 2.10, you may have a batch file execute a command

or not, depending on the truth or falsity of a particular condition. You may also specify looping of commands.

The enhanced features of DOS 2.10 batch commands are controlled by the subcommands:

```
ECHO
FOR
GOTO
IF [NOT]
SHIFT
PAUSE
REM
```

You use these commands within a batch file exactly as if they were DOS commands.

ECHO

The ECHO subcommand controls certain aspects of screen display during execution of a batch file. Under default conditions, DOS displays each DOS command as it is being performed. For example, suppose that a batch file is executing the command:

```
COPY *.* C:
```

Then DOS will display the command while it is being executed. The subcommand:

```
ECHO OFF
```

disables the command display. Note, however, that ECHO OFF does not affect displays other than the display of the current command.

The subcommand:

```
ECHO ON
```

cancels the ECHO OFF subcommand. This returns the command display to the default condition.

For example, here is a batch file employing ECHO:

```
ECHO OFF
COPY FILE1 C:
COPY FILE2 C:
COPY FILE3 C:
ECHO ON
```

The ECHO OFF turns off the display of subsequent commands. The ECHO ON returns DOS to the default condition in which commands are displayed. Note, however, that the command ECHO OFF is displayed and will remain displayed while the other commands of the batch file are executed. You may

remove the display of the command ECHO OFF by following that command with the DOS command CLS. This clears the screen, erasing ECHO OFF.

You may display a message from a batch file using the subcommand:

```
ECHO <message>
```

For example, to display the message “Copying Files...Wait”, you could use the command:

```
ECHO Copying Files...Wait
```

PAUSE

The PAUSE subcommand lets you pause execution of a batch file so you can take some action (such as to change diskettes). The PAUSE subcommand displays the prompt:

```
Strike a key when ready
```

Execution of the batch file is suspended until a key is pressed. You may include an optional remark with the PAUSE subcommand. For example, to display “Insert Data Diskette 2” and pause, we would use the command:

```
PAUSE Insert Data Diskette 2
```

The display will read:

```
Insert Data Diskette 2  
Strike a key when ready
```

TEST YOUR UNDERSTANDING 1 (answer on page 142)

Write a batch file that copies FILE1 onto FILE2. The batch file should pause and display the instruction “Insert diskette containing FILE1”.

REM

This subcommand lets you annotate a batch file with remarks. Note, however, that if ECHO is ON, then the remarks will be displayed as the batch file encounters the REM subcommands.

Labels and GOTO

You may insert labels to identify various points in a batch file. A label is a string of up to eight characters. (You may actually use more than that, but only the first eight are significant.) For example, here is a batch file in which we have included the label LABEL1.

```
LABEL1:  
COPY FILE1 FILE2
```


Note that the label has a colon after it. Labels are not displayed as commands.

The GOTO subcommand allows you to control the flow of execution of a batch file by causing a jump to a label. For example, to jump to LABEL1, we would use the command:

```
GOTO LABEL1
```

When this command is encountered in a batch file, DOS checks the file for the location of LABEL1. If LABEL1 is found, then execution resumes with the command immediately after LABEL1. If the label is not found, then DOS prints the message:

```
Label not found
```

and goes to the next command in the batch file.

TEST YOUR UNDERSTANDING 2 (answer on page 142)

Write a batch file to repeatedly display the volume label of the diskette in drive A:. (The only way to stop such a batch file is to use Ctrl-Break.)

FOR

The FOR subcommand lets you execute loops within a batch file. The format of this subcommand is:

```
FOR %%variable IN <set> DO <command>
```

Here %%variable is a dummy parameter that may be used within <command>. This dummy parameter must be preceded by %%. An example is %%FILE.

The dummy parameter %%variable is set equal to the values in <set>. For instance, to let %%FILE assume the values FILE1, FILE2 and FILE3, <set> would assume the form:

```
(FILE1 FILE2 FILE3)
```

Note that the items of <set> are in parentheses and individual items are separated by spaces.

The <command> may be any DOS command or a program that runs under DOS. The FOR subcommand allows %%variable to assume each value in <set>. It then performs <command>. For example, to copy FILE1, FILE2, and FILE3 onto the diskette in drive B:, we could use the command:

```
FOR %%FILE IN (FILE1 FILE2 FILE3) DO COPY %%FILE B:
```

At first %%FILE is equal to FILE1 and DOS performs the command:

```
COPY FILE1 B:
```

Next %%FILE is equal to FILE2 and DOS performs the command:

```
COPY FILE2 B:
```

Finally, %% FILE is equal to FILE3 and DOS performs the command:

```
COPY FILE3 B:
```

TEST YOUR UNDERSTANDING 3 (answer on page 142)

Use the FOR command to run the BASIC programs PROG1.BAS, PROG2.BAS, and PROG3.BAS one after another.

IF

The IF subcommand allows you to execute commands conditionally. The possible forms of the IF subcommand are:

```
IF <condition> <command>
IF NOT <condition> <command>
```

For the first form, if <condition> is true, then <command> is executed; if <condition> is false, the control skips to the next command in the batch file. For the second form, if <condition> is false, then <command> is executed; if <condition> is true, then control skips to the next command in the batch file.

<command> may be any DOS command or a program name.

There are three possible forms for <condition>:

EXIST filespec—This form allows you to test if the given file specification (filespec) actually corresponds to a file in the current directory. (Paths may not be included.)

string1 = string2—This form allows you to compare two strings, one or both of which may be given by dummy parameters specified when the batch file was invoked.

ERRORLEVEL number—This form allows you to test if the preceding DOS command ended with an ERRORLEVEL at least equal to <number>.

Let's give a few examples of IF commands using each type of condition. Here is a batch file to copy the FILE1 onto A:FILE2.

```
IF EXIST FILE1 COPY FILE1 A:FILE2
IF NOT EXIST FILE1 ECHO File Not Found
```

Note that the second IF statement prints an error message if FILE1 is not found.

In the above batch file, FILE1 and FILE2 were fixed. However, we may specify them as dummy parameters %FILE1 and %FILE2. In this case, the batch file would be:

```
IF EXIST FILE1 COPY %FILE1 A:%FILE2
IF NOT EXIST %FILE1 ECHO File Not Found
```

Suppose that the batch file is called COPYA. Then the batch file would be invoked by giving the command:

```
COPYA FILE1 FILE2
```

Here FILE1 is the file to be substituted for %FILE1 and FILE2 the file to be substituted for %FILE2.

Here is a command that checks whether two dummy parameters %1 and %2 are equal. If so, it prints out the error message "Incorrect input".

```
IF %1=%2 ECHO Incorrect input
```

DOS commands set a variable called ERRORLEVEL that indicates the level of completion achieved in executing the command. A value of ERRORLEVEL = 0 means that execution was successful. The various values for ERRORLEVEL are found in the DOS manual. For example, here is the listing of values for the command RESTORE:

- 0—Normal completion
- 1—No files found to restore
- 3—Terminated by user
- 4—Terminated due to error

You may, for example, use the IF command to determine if ERRORLEVEL is at least 1 (non-normal completion), in which case, you could repeat the command.

SHIFT

This command lets you use more than 10 dummy parameters. SHIFT causes dummy parameter 1 to become dummy parameter 0, parameter 2 to become dummy parameter 1, and so forth. To use more than 10 dummy parameters, just type the desired parameter values on the command line and use SHIFT to get at the values past the tenth.

ANSWERS TO TEST YOUR UNDERSTANDING

- 1: PAUSE Insert diskette containing FILE1
COPY FILE1 FILE2
- 2: LABEL1:

```
ECHO OFF
VOL
GOTO LABEL1
3: FOR PROG% IN (PROG1.BAS PROG2.BAS PROG3.BAS) DO
  BASIC PROG%
```

10.4 Redirecting Input and Output

In its default state, DOS receives its input from the keyboard and its output is sent to the display. However, you may tell DOS to redirect the input or output, replacing the standard devices with either a file or another device.

Redirecting Output

To a File—To send output to a file, you use a DOS command that begins with either > or >>. For example, to send output to the file MIDWEST\INVENT\MEMO.528, use either of the commands:

```
>MIDWEST\INVENT\MEMO.528
>>MIDWEST\INVENT\MEMO.528
```

The two commands differ only in case the indicated file currently exists. The first command replaces the existing file, while the second adds to it.

The above commands may be used in conjunction with DOS commands or user programs. For example, here is a command that runs the user program PROG1 and writes the output to the file A:OUTPUT.

```
PROG1 >A:OUTPUT
```

(If A:OUTPUT exists, it will be replaced with the output of PROG1.) Here is a command that writes the tree structure onto the file A:TREE.

```
TREE >A:TREE
```

Here is a batch file that runs PROG1 and PROG2 and accumulates the output from the two programs PROG1 and PROG2 in the file A:COMBINED.

```
PROG1 >A:COMBINED
PROG2 >>A:COMBINED
```

Note that the second command appends to the file A:COMBINED, which is started by the first command.

Redirection of output lasts only for the duration of the command or program. At the completion of the command or program, output is redirected back to the screen.

TEST YOUR UNDERSTANDING 1 (answer on page 144)

Write a command that:

- *runs PROG1 and writes its output in the file TEST1
- *runs PROG2 and writes its output on the display

Just as you may redirect output to a file, you may redirect output to a device. Just use a device name instead of a file name. For example, to redirect the output of PROG1 to the printer, you may use the command:

```
PROG1 'PRN:
```

TEST YOUR UNDERSTANDING 2 (answer on page 144)

Write a command to write the contents of the current directory to the file DRECTRY.CUR .

Redirecting Input

You may redirect a program or DOS command to receive its input from a file. For example, to run the program PROG1 with input from the file A:DIRECT, we would use the command:

```
PROG1 <A:DIRECT
```

Redirecting input is trickier than redirecting output. The file used for input must contain all the necessary program input. If you reach the end of the input file before all input needs are met, your program will wait indefinitely for the added input. If this happens, it will be necessary for you to return to DOS by pressing Ctrl-Break.

ANSWERS TO TEST YOUR UNDERSTANDING

- 1: PROG1 >TEST1
PROG2
- 2: DIR >DRECTRY.CUR

10.5 DOS 2.1 COMMAND SUMMARY

Command	Explanation	Format
ASSIGN	Reroute disk commands	ASSIGN [olddrive= newdrive[...]]

BACKUP	Create a backup of hard disk files	BACKUP [d:][path][filespec] [d:][S][M] [/A][D:mm-dd-yy]
BREAK	Enable check for Ctrl-Break	BREAK [ON-OFF]
CHDIR or CD	Change directory	CHDIR [[d:] path]
CHKDSK	Check status of disk	CHKDSK [d:][filename][F][V]
CLS	Clear screen	CLS
COMP	Compare files for equality	COMP [d:][path][filespec] [d:][path][filespec]
COPY	Copy files	COPY [/A][B][d:][path][filespec] [d:][path][filespec] [/A][B][V]
DATE	Enter date	DATE [mm-dd-yy]
DIR	Display directory	DIR [d:][filespec][P][W]
DISKCOMP	Compare two diskettes for equality	DISKCOMP [d:] [d:][1][8]
DISKCOPY	Copy one diskette onto another	DISKCOPY [d:] [d:][1]
ERASE or DEL	Delete file	ERASE [d:][path][filespec]
FORMAT	Format a diskette	FORMAT [d:][S][1][8][V][B]
GRAPHICS	Print screen in graphics mode	GRAPHICS
MODE	Set mode of printer Set mode of screen Set mode of communications adapter	MODE LPT#:[n][,][m][,][P] MODE [n],m[,T] MODE COMn:baud[,parity[,databits[,stopbits[,P]]]] MODE LPT#:=COMn
	Redirect output	
PATH	Define directory search paths	PATH[d:]path[;[d:]path...]
PRINT	Queue files for printing	PRINT[[d:][filespec][T][C][P]...]
RECOVER	Recover erased files	RECOVER [d:][path][filespec]
RENAME or REN	Rename a file	RENAME [d:][path]filespec newname
RESTORE	Restore files from a backup copy	RESTORE d:[d:][path][filespec][S][P]

RMDIR or RD	Delete a directory	RMDIR [d:]path
SYS	Copy DOS onto a diskette	SYS d:
TIME	Specify the current time	TIME [hh:mm:ss.xx]
TREE	Display the current tree structure of a directory	TREE [d:][[/F]
TYPE	Display the contents of a file	TYPE [d:][path]filespec
VER	Display the DOS version number	VER
VERIFY	Enable or disable verification	VERIFY [ON\$-OFF]
VOL	Display the volume label	VOL [d:]

APPENDIX: HARDWARE SPECIFICATIONS

1. SYSTEM UNIT

Central Processing Unit: Intel 8088 Microprocessor
Memory

Access time 250 nanoseconds

Cycle time 375 nanoseconds

64k Internal ROM containing diagnostics, cassette BASIC, keyboard tutorial, BIOS, cassette operating system, character generator for graphics mode, diskette bootstrap loader

Program cartridge slots can accommodate Cartridge BASIC and/or cartridge programs in ROM cartridges

64k RAM standard

Additional 64k RAM optional

All memory 8 bits wide, not parity checked

Standard System Unit Interfaces and Connectors

Transformer Connector

Serial Interface

Cassette Interface

Joystick Interfaces

Keyboard Interface

cordless

with cord

Modem Interface

Diskette Interface

Display Interface

direct drive video

composite video

television

Light Pen Interface

3 expansion slots (dedicated to modem, memory/display, and diskette)

3-voice audio output

Physical Data

Dimensions

13.9 inches long x 11.4 inches deep x 3.8 inches high

Weight

Basic unit approximately 6 pounds

Enhanced unit approximately 9 pounds

Electrical requirements

120v 60Hz

2. KEYBOARD

Description

Can be operated without a cord via an infrared optical link or connected to the system unit via an optional 6 foot flat cord. Optical link allows operation up to 20 feet from the system unit, provided there are no line-of-sight obstructions or other computer systems operating in the immediate vicinity.

When using optical link, requires 4 AA batteries.

Keyboard has legs that position keyboard at an angle of 9 degrees to the horizontal.

Dimensions

13.45 inches x 6.61 inches x 1.02 inches

Weight

22 ounces without batteries

25 ounces with batteries

3. TRANSFORMER

Description

Transforms receptacle current into a form suitable for the system unit. Housed in a separate desktop unit that connects to the system unit via a cable and connects to a standard 110 volt outlet via a grounded plug.

Operating Temperature Range

60 to 90 degrees Fahrenheit

Operating Humidity Range

8% to 80%

Altitude Limitation

Up to 7000 feet

Weight

2.8 pounds

4. PCjr DISKETTE DRIVE

Description

A slim line 5¼-inch diskette drive. Capable of reading and writing single- and double-sided double-density diskettes. Each diskette side has 40 tracks with nine 512-byte sectors, for a total diskette capacity of 180k per side. The diskette drive is controlled by the diskette adapter board, which is mounted in the appropriate expansion slot in the main system board. The diskette drive mounts in a compartment at the top front of the system unit. It may be added as an option to the basic unit, but comes standard with the enhanced unit.

Speed

300 RPM

6 milliseconds track-to-track access time

250k bits/second data transfer rate

Operating Altitude

Up to 7000 feet

Weight

2.2 pounds

Dimensions

1.6 inches x 5.8 inches x 8.3 inches

5. IBM COMPACT PRINTER**General Characteristics**

Thermal print head

Requires special thermal paper, available in either cut sheet, roll, or fanfold

50 characters per second

Compact, light, and quiet

Printing Statistics

Maximum print line: 8.5 inches

Standard Character size: 10/inch

Alternate print mode:

Double width : 5 characters per inch

Compressed : 17.5 characters per inch

Double width and compressed : 8.75 characters per inch

Graphics : all dots addressable

70 dots horizontally/inch

8 dots vertically per 1/9-inch line

Vertical text spacing : 6 or 9 lines per inch

Font : 191 characters including all 128 ASCII characters

Physical Characteristics

Plugs into PCjr System Unit

Uses modified RS232-C interface

Weight : 6.6 pounds

Dimensions : 12.3 inches x 8.675 inches x 3.5 inches

Noise Generated : 55 decibels

Power Supply : 110 volts

6. PCjr INTERNAL MODEM

Description

Answer or originate transmission manually or automatically
Can be used with either rotary or push-button phone line
100/300 baud transmission rate
Plugs into expansion slot of PCjr; does not require RS232-C interface
Bell series 103 compatible
Power-on self-test
Error-detection capability
Fully programmable from PCjr BASIC
 7- or 8-bit characters
 Parity: even, odd, or none
 Stop bits: 1
Attaches to standard modular phone jack.

ANSWERS TO EXERCISES

CHAPTER 9

Section 9.1 (page 101)

1. Valid
2. Valid
3. > is an illegal character
4. The colon is an illegal character
5. This is legal (believe it or not!)
6. Use only one period
7. Extensions may only have 3 characters.
8. Names may only have 8 characters.

Section 9.2 (page 104)

- | | |
|-----------------|-----------------|
| 1. COM1:ALICE.3 | 2. LPT1:MESSAGE |
| 3. A:*. * | 4. *. * |
| 5. A:?. * | 6. A:RALPH.* |

Section 9.4 (page 108)

1. COPY B:TEST LPT1:
2. COPY A:*.COM B:
3. ERASE A:TEST
4. COPY A:TEST B:TEST3
5. COPY A:D??????.* B:

INDEX

- 1-2-3, 77
- Acoustic modem, 35
- Adventure in Serenia, 57
- Adventures in Math, 59
- Animation, 40
- Animation Creation, 57
- Answers to exercises, 152
- Arcade-style game, 8
- Arithmetic, 16
- Arithmetic Games, 59
- ASSIGN, 132
- Asynchronous communications adapter, 131
- AUTOEXEC.BAT, 117

- Backslash, 122
- Backup, 92, 132
- Banking, 8
- Bar chart, 18
- BASIC, 1, 78
- BASIC Compiler, 64
- BASIC Programming Development System, 64
- BASIC prompt, 89, 90
- Batch file, 115, 137
- Baud, 35
- Bell-shaped curve, 19, 20
- BIOS call, 80
- BREAK, 133
- Budget records, 8
- Bulletin board, 17
- Bumble Games, 59
- Bumble Plot, 59
- Bytes, 3

- Cable, 28
- Caps lock key, 96
- Cartridge BASIC, 37, 38
- Cartridge game, 8
- Casino Games, 58
- Cassette, 2
- Cassette BASIC, 37
- Cassette recorder, 3, 5, 31
- Cautions in handling diskettes, 87
- Central processing unit (CPU), 2
- Charts, 16
- CHKDSK, 115, 130
- CIRCLE, 40
- Circuit board, 28
- CLS, 133
- Colors, 39
- Communications, 17, 34, 46, 65
- COMP, 114, 129
- Compatibility of software, 79
- Computer art, 19, 20
- COPY, 106, 108, 109, 123
- Copying a diskette, 92
- Crossfire, 59
- Current directory, 122
- Cursor, 90

- Data file, 99
- DATE, 113, 128
- Device name, 102
- DIR, 129
- Direct memory access, 80
- Direct-connect modem, 35

- Directory, 101, 121
- Disk operating system, 1, 88
- DISKCOMP, 114, 129
- DISKCOPY, 92, 129
- Diskette, 2, 85
- Diskette drive, 3, 4, 5, 6, 24, 32, 77, 85, 91, 149
- Diskette drive adapter, 29, 32
- Display, 2, 25, 32
- Display adapter, 30, 33
- DOS, 1, 38, 56, 85
- DOS command summary, 144
- DOS prompt, 89
- DOS Supplementary Programs, 88
- DOS Technical Reference Manual, 56
- Dow Jones News Service, 58
- Dow Jones Reporter, 60

- EasyWriter, 16, 60
- ECHO, 138
- Education, 60
- ENTER, 89
- Entertainment, 59
- ERASE, 113
- Event trapping, 46
- Executing commands, 104
- Executing programs, 104
- Expansion slots, 29, 30, 78
- External DOS command, 105
- External modem, 35

- File, 99
- File name, 99
- File specification, 102, 103
- Finance, 61
- Financial decision-making, 9
- Floppy diskette, 4
- FOR, 140
- FORMAT, 109, 110, 130
- Formatting commands, 14

- Geography, 16
- GET, 40, 46
- GOTO, 139
- GRAPHICS, 133
- Graphics Macro Language, 38, 46
- Graphics mode, 32, 33, 39
- Graphs, 16, 20

- Hands-On BASIC, 17, 46
- Hard copy, 2
- Hard disk, 77
- Hardware specifications, 147
- Home Budget jr, 60
- Home computer, 1
- Home finance, 8
- Home video system, 8
- Home Word, 16, 61

- IBM 80 cps Dot Matrix Printer, 34
- IBM color display, 32, 33
- IBM Compact Printer, 33, 150
- IBM Graphics Printer, 34
- IBM publications, 64
- IBM software, 59
- IF, 141

- Infrared optical link, 23
- Input unit, 2
- Internal DOS command, 105
- Internal modem, 30, 35, 151
- Interpreted language, 37
- Investing, 9
- Joystick, 26
- Juggles' Butterfly, 58
- Keyboard, 6, 25, 148
- Labels, 139
- Learning DOS, 17
- Light pen, 27
- LINE, 40
- LOGO, 59
- Macro Assembler, 65
- Master DOS diskette, 88
- Memory, 2, 77, 80
- Microprocessor, 2, 28
- Mine Shaft, 58
- MODE, 130
- Modem, 28, 34
- Monitor, 32, 33, 78
- Monster Math, 60
- Mortgage, 10
- Mouser, 58
- MS-DOS, 88
- Multiplan, 60
- Music Macro Language, 20, 38, 46
- Music synthesizer chip, 8, 20
- Operating system, 88
- Output unit, 2
- PAINT, 40
- Parallel interface, 34
- Parameters of a batch file, 118
- Parity checking, 77
- PATH, 134
- PAUSE, 139
- PC, 2, 77
- PC Junior, 1
- PC-DOS, 88
- PC/XT, 2, 77
- PCjr, 5
- PCjr BASIC, 79
- PCjr basic system, 5
- PCjr enhanced system, 6
- PCjr Internal Modem, 35
- PCjr BASIC, summary of, 47
- PeachText, 61
- Personal Communications Manager, 65
- Personal Editor, 65
- pfs:FILE, 61
- pfs:REPORT, 61
- Pie chart, 18
- Planning a PCjr system, 81
- Plotter, 20
- PRINT, 134
- Printer, 2, 20, 25, 130
- Professional Editor, 65
- Program, 99
- Programming, 64
- PUT, 46
- RAM, 2, 3, 6, 29
- Read-write window, 86
- RECOVER, 135
- Redirecting input, 144
- Redirecting output, 143
- Redirection, 131
- REM, 139
- RENAME, 113, 129
- Reset, 96
- RESTORE, 135
- RMDIR, 127
- ROM, 2, 2, 29
- Root directory, 122
- Screen, 131
- Screen width, 80
- Scrolling, 96
- Scuba Venture, 58
- SHIFT, 142
- Shift key, 96
- Soft-sectored diskette, 110
- Software compatibility, 79
- Source, 17
- Specifications, 147
- Spreadsheet, 10
- Starting your computer, 88
- Strategy game, 8
- Strategy Games, 58
- Subdirectory, 121
- Summary of PCjr BASIC, 46
- SYS, 128
- System configurations, 65
- System costs, 72
- System unit, 6, 24, 147
- Tax planning, 9
- Tax returns, 9
- Television set, 32
- Terminal, 17
- Thermal printer, 33
- TIME, 114, 128
- Time and data management, 64
- Time Manager, 61
- Tracks, 109
- Transformer, 6, 25, 149
- TREE, 136
- Turtle Power, 60
- TYPE, 114, 130
- Typing, 16
- User's guide, 2
- VER, 136
- VERIFY, 136
- Video monitor, 2
- VIEW, 40
- VisiCalc, 10, 60
- VOL, 137
- Volkswriter, 16
- Wild card character, 103, 108
- WINDOW, 40
- Word processing, 14, 63
- Word Proof, 61
- WordStar, 16
- World coordinates, 40
- Write-protect notch, 86

Now! The book that explains why the IBM PCjr is today's standard for personal home computing!

IBM PCjr Buyer and User Guide

Larry Joel Goldstein

If you're thinking about purchasing a home computer system, here's the book that shows you everything you need to know about the IBM PCjr! This comprehensive book takes you on a start to finish tour of what the IBM PCjr is ... what it can do ... and how it can meet your particular needs.

You'll learn how an IBM PCjr is composed of a number of separate components from which you can choose to configure a system that works best for you. In fact, this guide literally walks you through the entire system selection and purchasing process—from your first thoughts on purchasing a new system till the time you get home to set up and use your new IBM PCjr!

In this book you will find:

- Complete descriptions of the IBM PCjr's system components
- Typical system configurations
- Important software considerations

CONTENTS

A First Look at Computers/What Can You Do With the PCjr?/A Close Look at the PCjr's Hardware/Software & Publications for Your PCjr/Typical System Configurations/Other Members of the IBM Family/PCjr System Planning Aids/Using DOS for the First Time/An Introduction to DOS/More About DOS/Appendix: Hardware Specifications/Answers to Exercises/Index

ISBN 0-89303-546-7