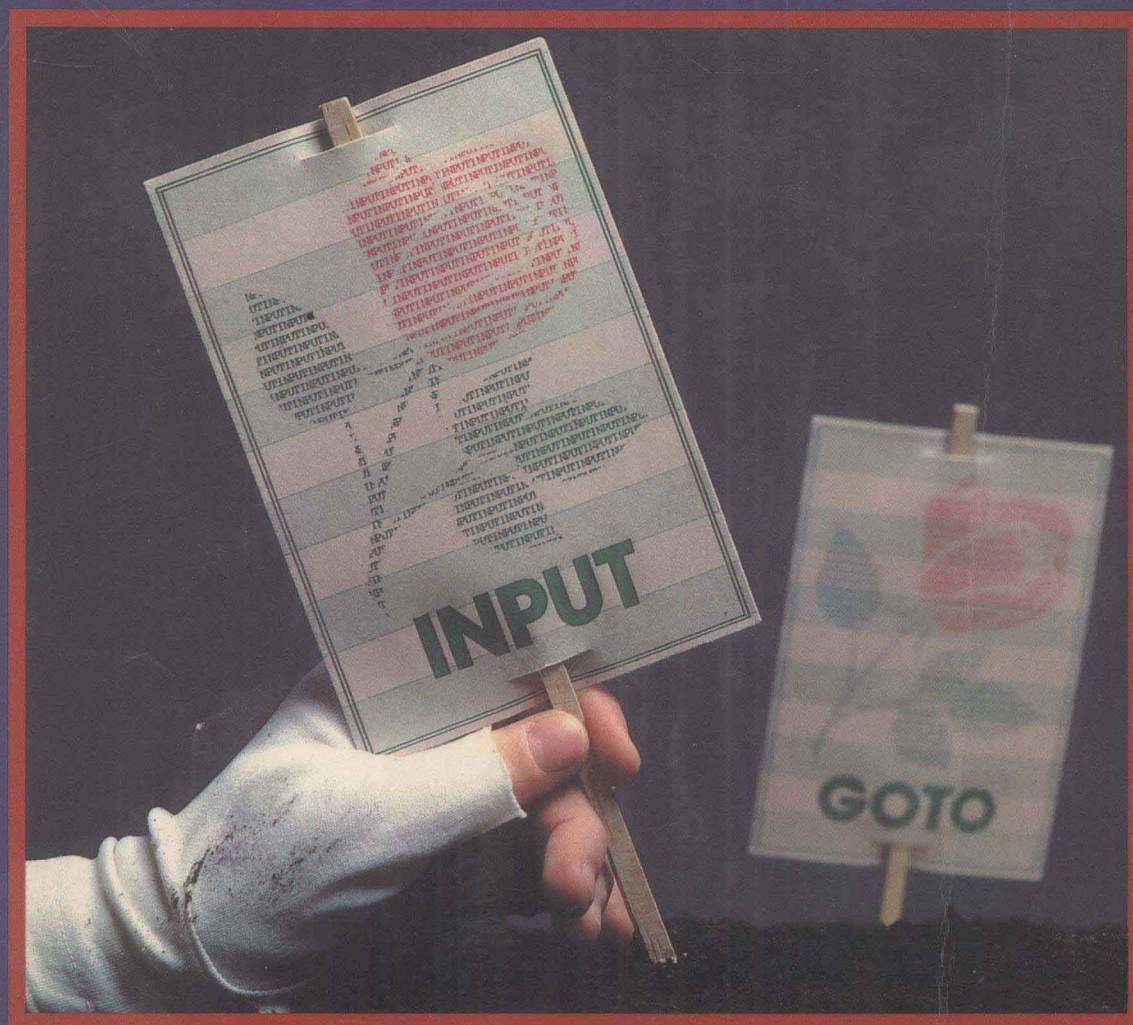
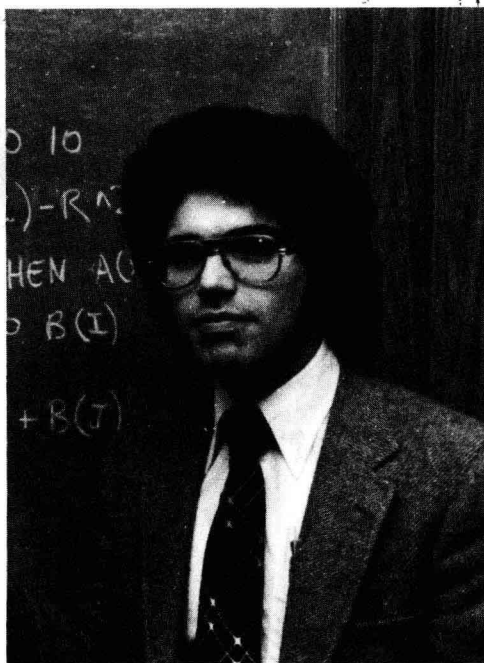


# BASIC Programming With the IBM PCjr™

David C. Willen



# **BASIC Programming With the IBM PC*jr*<sup>TM</sup>**



**David C. Willen** is a recognized authority in the microcomputer field. He is co-founder and currently president of CAU (Computer Applications Unlimited), Inc., a firm specializing in the development and marketing of state-of-art microcomputer software. He is in constant demand as a writer, consultant and speaker—while his widely acclaimed book: *8088 Assembler Language Programming: The IBM PC* has established a high water mark in the technological book arena. Mr. Willen has a keen interest in classic and operatic music and is an avid amateur radio enthusiast (WB2DZO).

# **BASIC Programming With the IBM PCjr<sup>TM</sup>**

by

**David C. Willen**

**Howard W. Sams & Co., Inc.**  
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

Copyright © 1984 by David C. Willen

FIRST EDITION  
FIRST PRINTING—1984

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-672-22359-7  
Library of Congress Catalog Card Number: 84-50284

Edited by *Phil Lieberman*

Printed in the United States of America.

## PREFACE

The arrival of an IBM home computer has been anxiously awaited for some time. This new machine, the PCjr, provides far greater power and capabilities than previous home computers. Furthermore, IBM's commitment to maintaining compatibility with its larger systems—in this case, the PC—insures that a large and electric library of software will be available for the PCjr. For most people thinking about buying a home computer, these reasons make the PCjr an excellent choice.

Once the machine is in the home, the new user will be busy trying out the various software packages that were purchased with it. The delight of playing sophisticated and educational games will soon be followed by the more industrious effort of using personal productivity software such as the *Home Budget Manager*. Eventually, the desire to write a program asserts itself.

This book will help you learn how to write your own programs for the PCjr. The BASIC programming language that is provided with the machine is powerful and complex. The documentation that accompanies it is therefore also rather lengthy. This book is not a reference work—it is meant to be read straight through, preferably with a PCjr at your side. As each fundamental concept and capability of PCjr BASIC is presented, the emphasis will be on practical applications and programming *technique*.

To illustrate these various techniques, many sample programs are provided. Several of the larger programs will appear repeatedly throughout the book, expanding and becoming more powerful as new capabilities are learned and added. One of the great joys of computer programming is the discovery of new and better ways to accomplish a specific task—this acts as a strong inducement to continue one's study of the subject. As more and more programming commands are learned, there is always a "better way to do it" around the corner.

David C. Willen



# TABLE OF CONTENTS

CHAPTER 1	
INTRODUCTION .....	11
A Look Inside • Main Memory • The BASIC Interpreter • Your First Program	
CHAPTER 2	
THE BASICS OF BASIC .....	17
The Elements of a Program • Keywords • Variables • Expressions	
The Command Interpreter • Stopping a Program—Break	
Viewing the Program—List • Program Editing	
Inserting and Deleting Program Text • Program Renumbering	
Program Filing • Immediate Results • The Function Key	
CHAPTER 3	
INPUT AND OUTPUT .....	33
Displaying Results with PRINT • The TAB Function	
The PRINT USING Function • Controlling the Appearance of the Display	
Positioning the Cursor • Controlling Color	
Sending Output to the Printer • The INPUT Statement Revisited	
Storing Data Within a Program • A Colorful Example	
CHAPTER 4	
PROGRAM CONTROL AND LOOPS .....	51
The IF-THEN Statement • The ELSE Clause • The FOR-NEXT Loop	
Variable Range Loops • The STEP Clause	
Nested FOR-NEXT Loops • The ON-GOTO Statement	
CHAPTER 5	
CHARACTER STRINGS .....	67
Character String Variables • A Simple Example • String Comparisons	
The ASCII Standard • The Value of a Standard • String Concatenation	
Character String Functions • The MID\$ Statement	
Functions That Yield Numeric Results • Conversion Functions	
The Secret Code Program • The Square Drawing Program	
More Input Techniques • The INKEY\$ Function	
The Secret Password Program	



CHAPTER 6	
ARRAYS AND FUNCTIONS .....	89
The Array Concept • The DIM Statement • The Importance of the Subscript	
Multidimensional Arrays • A Home-Database	
Improving the Database Program • How to Sort Information	
Mathematical Functions • Logarithmic Functions • Trigonometric Functions	
The Random Functions	
CHAPTER 7	
SUBROUTINES AND USER-DEFINED FUNCTIONS .....	121
The Basic Subroutine • Multiple Subroutines	
Subroutine Parameters and Modular Programming • User-Defined Functions	
The DEF-FN Statement • Nested Function Definitions	
Multiple Argument Functions • Program Variable References	
CHAPTER 8	
FILE I/O .....	139
The File Concept • Filenames • Types of File Access • File Access Protocol	
File Numbers • Sequential Files • Writing Sequential Files	
Reading Sequential Files • File Structure	
Enhancing the Home-Database Program • Random Files	
Defining Random Files • Accessing Random File Data	
The LSET and RSET Statements • A Sample Program	
CHAPTER 9	
CREATING SPECIAL EFFECTS .....	163
Generating Sound and Music • The SOUND Statement	
The NOISE Statement • Mixing Noise with other Sounds	
The PLAY Statement • Setting the Length of Notes	
Programming Music Rests • Controlling Tempo • Multiple Voices	
Graphics Display Modes • Graphics Screen Layout • Color Attributes	
Color Palettes • Creating Graphics Images • The LINE Statement	
The CIRCLE Statement • The PAINT Statement • A Pie Chart Program	
Advanced Palette Control	
Appendix.....	191
Index.....	195

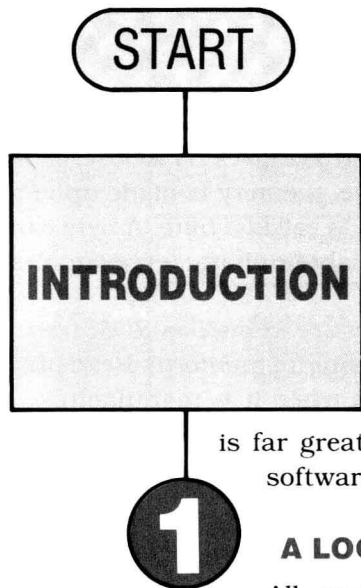
## **ACKNOWLEDGMENTS**

I would like to thank the IBM Corporation for its assistance during the preparation of this book. In particular, the sales people at the various IBM Product Centers that I visited provided invaluable information and guidance regarding their newest computer.

I would also like to thank Bill Oliphant and the entire staff at Howard W. Sams for their assistance in producing the completed product that you are now holding.

## **CHAPTER ONE**

**A Look Inside • Main Memory  
The BASIC Interpreter • Your First Program**



The IBM PCjr is a remarkable little computer, and programming it can be both enjoyable and rewarding. In this book, we will learn about the PCjr and how to program it with the BASIC programming language. By writing your own programs, you will gain a feeling of confidence and mastery over the computer. The degree of control over the machine that you will achieve

is far greater than can be realized when using packaged software.

# 1

## A LOOK INSIDE

All good craftsmen have a thorough knowledge of the tools and equipment with which they work. A good programmer, therefore, must have a solid understanding of the computer's hardware. He or she must also be familiar with the various *software tools* that are used to create and modify programs. Let us start by looking at a *block diagram* of our computer-of-choice, the PCjr.

In Fig. 1-1, we can see that the PCjr consists of a *central processing unit* (CPU) which is attached to *main memory* as well as several *input/output devices*. The CPU is the "brains" of the computer. Whenever the computer performs a calculation or makes a decision, the CPU is used. *Programming* the computer consists of giving the CPU a sequence of instructions to follow. While the CPU is *executing* these instructions, they are stored in main memory. Main memory is also used to store data that is to be processed by the program. The input/output devices provide the computer with access to the "outside" world (keyboard and video display). Some input/output devices are used for long term storage of data (cassette, disk).

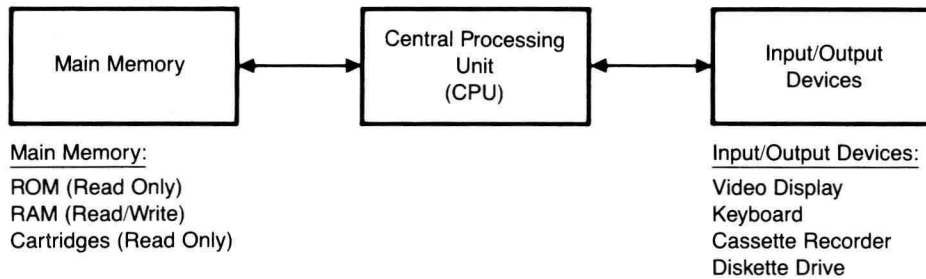


Fig. 1-1. Block diagram of the PCjr.

## MAIN MEMORY

As programmers, main memory is always very important to us, because it is where our programs and data will reside. Fig. 1-2 gives us a closer look at a section of computer memory. As you can see, memory is made up of a number of data storage cells. Each of these cells is called a *byte*. A byte can be used to store either a small number or an alphabetic character; examples of both are shown in Fig. 1-2.

The PCjr utilizes two kinds of memory; they are known as *ROM* (*read only memory*) and *RAM* (*random access, or read/write memory*). Read only memory is set to contain a fixed pattern of data when it is manufactured.

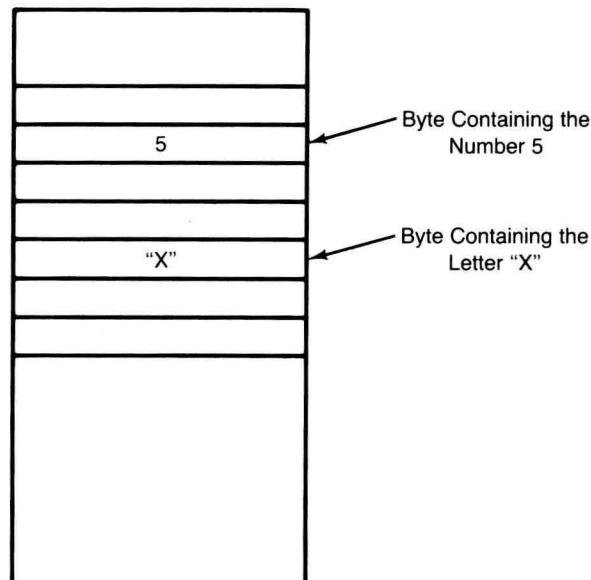
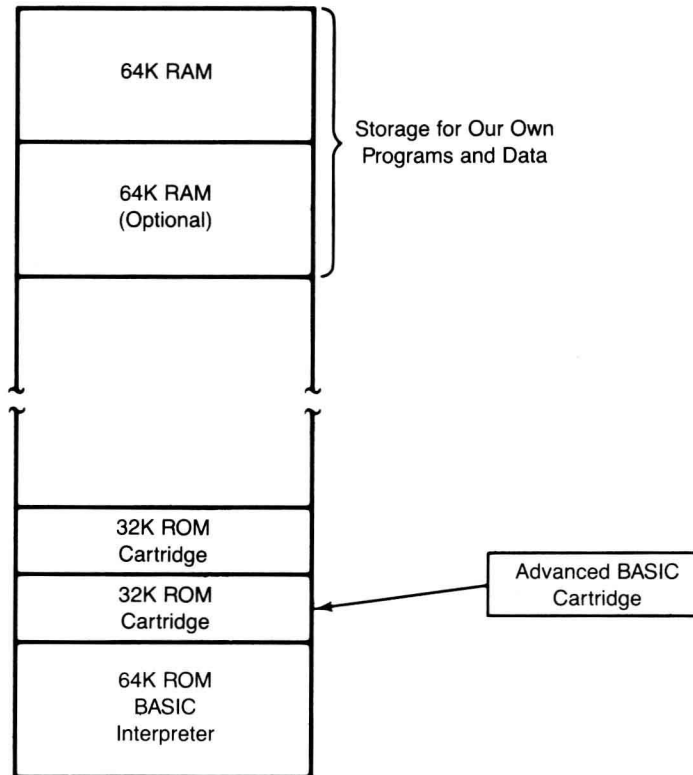


Fig. 1-2. Main memory.



**Fig. 1-3.** PCjr Memory map.

The data contained in ROM can be *read*, but cannot be altered. ROM is therefore used to store fixed programs such as the *BASIC interpreter* and other software packages available on ROM cartridges. For the purposes of storing our own programs and data, we can use RAM. We can store data in RAM which can be retrieved at a later time. As long as the computer is turned on, the RAM will retain our information.

The *memory size* of a computer is a measure of how much data the computer can store in its main memory. The unit of measurement is a byte, and the value is usually presented with the letter "K" as a suffix. The "K" means multiply by 1024. So when you tell someone that your PCjr has 64K RAM, you are saying that the machine has a memory capacity of  $64 \times 1024$ , or 65,536 bytes.

Of course, your PCjr has a much greater memory capacity when you take into account its ROM. Fig. 1-3 shows a *memory map* of the PCjr. Here we can see how the ROM and RAM are organized within the computer. Note that we can expand the read/write memory of the PCjr to a maximum of 128K by adding an additional 64K RAM option. Additional ROM can be

added by plugging cartridges into one or both of the PCjr's cartridge slots. Each cartridge can contain up to 32K of ROM.

## THE BASIC INTERPRETER

The *BASIC interpreter* is the program that provides us with the ability to create and run our own BASIC programs. Since it is a fixed program, it is stored in the ROM of the PCjr, as seen in Fig. 1-3. By adding the additional ROM contained in the optional Advanced BASIC cartridge, we can enhance the BASIC interpreter. There are more commands and features available in Advanced BASIC than in the standard BASIC. Many of the sample programs in this book do not require the Advanced BASIC cartridge, however, those that do are so marked.

You might ask, "If the BASIC interpreter provides me with the ability to run my BASIC program, what provides the ability to run the BASIC interpreter itself? After all, you have told me that the interpreter is a program; and I have seen that it resides in memory just like my own program would."

A very good question, and the answer is that the CPU (Fig. 1-1) runs the BASIC interpreter program. This is possible because the interpreter is an assembly language program. "Oh, no!," you cry, "I've heard of assembly language—it's that mysterious and frightening language that nobody understands!" Do not be afraid, you do not need to learn anything about assembly language to read this book and become a good BASIC programmer. It does not hurt, however, to be aware of assembly language and its role in the PCjr system.

## YOUR FIRST PROGRAM

Our first sample program will get us off to a good start by being both simple and useful. The program listing is given in Fig. 1-4. If you have ever used an adding machine or four function calculator to keep a "running total" as you balance your checkbook, you will appreciate this program. It basically provides you with the same ability. To use this program, turn on your PCjr. When you get the "Ok" prompt from BASIC, type in each program line shown in Fig. 1-4. Press the ENTER key at the end of each program line. When you have typed in the entire program, start it by typing RUN and pressing ENTER.

```
10 REM ADDING MACHINE PROGRAM
20 LET TOTAL = 0
30 INPUT X
40 LET TOTAL = TOTAL + X
50 PRINT TOTAL
60 GOTO 30
```

**Fig. 1-4.** An adding machine program.

```
OK
RUN <ENTER>
? 100 <ENTER>
  100
? 250 <ENTER>
  350
? 100 <ENTER>
  450
? -50 <ENTER>
  400
? -25 <ENTER>
  375
? -200 <ENTER>
  175
? <BREAK>
Break in 30
OK
```

**Fig. 1-5.** Sample run of adding machine program.

Your PCjr will display a question mark (" ? ") and await your input. For each deposit, type in the dollar amount and press ENTER. Checks and other withdrawals can be entered by preceding the dollar amount with the minus sign (" - "). The sample run in Fig. 1-5 shows deposits of \$100, \$250, and \$100, followed by withdrawals of \$50, \$25, and \$200. Notice how the program keeps track of the running total and displays it after each entry.

Congratulations! You are running your first BASIC program on the PCjr!



## **CHAPTER TWO**

**The Elements of a Program • Keywords**

**Variables • Expressions**

**The Command Interpreter**

**Stopping a Program—Break**

**Viewing the Program—List • Program Editing**

**Inserting and Deleting Program Text**

**Program Renumbering • Program Filing**

**Immediate Results • The Function Key**