



**VAX<sup>®</sup>**  
**BASIC**  
for  
**BUSINESS**

**A Screen-based Approach**



**Nicholas A J Hastings**  
**Robert J Willis**

# **VAX BASIC for Business**

## **A Screen-based Approach**

Nicholas A. J. Hastings

Robert J. Willis

*Faculty of Economics and Politics  
Monash University*



**Prentice-Hall of Australia Pty Ltd**

© 1984 by Prentice-Hall of Australia Pty Ltd. All rights reserved. No part of this book may be reproduced in any form or by any means without permission in writing from the publisher. The program material contained herein or in any further deletions, addenda, or corrigenda to this manual or associated manuals or software is supplied without representation or guarantee of any kind. These computer programs have been developed for student use in a teaching situation and neither the authors nor Prentice-Hall of Australia Pty Ltd assume any responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of these programs or part thereof.

Prentice-Hall of Australia Pty Ltd, Sydney  
Prentice-Hall International Inc., London  
Prentice-Hall Canada Inc., Toronto  
Prentice-Hall of India Private Ltd, New Delhi  
Prentice-Hall of Japan Inc., Tokyo  
Prentice-Hall of Southeast Asia Pte Ltd, Singapore  
Editora Prentice-Hall do Brasil LTDA., Rio de Janeiro  
Whitehall Books Ltd, Wellington  
Prentice-Hall Inc., Englewood Cliffs, New Jersey

2 3 4 5 87 86 85 84

Printed and bound in Australia by  
Impact Printing Pty Ltd, Brunswick, Victoria

Cover by Joel Solomon

VAX, DEC and VT are registered trademarks of  
Digital Equipment Corporation, Maynard, Mass.

U.S. ISBN: 0-13-940941-6

National Library of Australia  
Cataloguing-in-Publication Data

Hastings, N.A.J. (Nicholas Anthony John), 1937- .  
VAX BASIC for business.

Includes index.  
ISBN 0 7248 1253 9.

1. VAX-11 (Computer) - Programming.
2. BASIC (Computer program language).
3. Business - Data Processing. I. Willis, Robert J. (Robert John), 1950- . II. Title.

001.64'24

# VAX BASIC for Business

A Screen-based Approach

# Preface

The aim of this book is to explain how to write BASIC programs for business applications, making use of the features of the video screen terminal. No previous knowledge of computing is assumed.

Video screens are now universally used as the human interface of computer systems. Despite this, existing texts on computer programming deal almost exclusively in programming techniques which are appropriate to the punched card era. This book presents a screen-based approach to programming for the modern computer user.

Screen-based programs are easy to write and very easy to use. Their main characteristic is that they guide the user through the application to which they relate, in a simple and self-explanatory way.

We start by introducing the elements of the BASIC programming language and give examples in the form of simple programs. Then we show how to write programs which use the video screen, thus making it easy for the user to operate the program. Gradually we build up a full explanation of how to write programs in BASIC with such features as:

Menus

Data entry screens

Screen based enquiry reports

Systems containing many different screens

Business data storage and retrieval

Printed reports

Finally we look at program design and explain a simple approach to developing complex programs in a modular fashion.

The programs in this book have been developed and tested on a VAX computer, with some additional programs for

microcomputers. In regard to cursor control, we give details of the techniques appropriate for VAX, IBM-PC, Apple, TRS-80 and Commodore computers. The Apple commands refer to Applesoft Basic and the TRS-80 commands refer to compiler Basic for the Model II and Model 16. Details are given of the cursor addressing sequences specified by the American National Standards Institute (ANSI). These sequences are used by DEC VT100 Series terminals, which are the terminals most frequently used with VAX computers. Most other computers use one of the above standards, or use similar procedures to which the user can readily adapt.

We would like to express our thanks to Dorothy Jones who prepared the text in camera ready form, and to Ng Soo San and Ying Wan Kong who assisted in the program checking and provided solutions to many of the exercises.

# Contents

<b>Preface</b>	ix
<b>1 Elements of BASIC</b>	1
1.1 What is a Computer?	1
1.2 What is a Program?	1
1.3 Components of a Computer	2
1.4 Creating, Storing and Running a Program	3
1.5 Errors and Debugging	6
1.6 The PRINT Statement and the END Statement	7
1.7 String Variables	8
1.8 Remark Statements	10
1.9 Numeric Variables	11
1.10 The INPUT Statement	13
1.11 Data Entry With Text Prompts	15
1.12 Joining String Variables Together	15
1.13 Positioning Output Within a Line	16
1.14 The TAB Function	18
1.15 RESEQUENCE	20
1.16 Exercises	21
<b>2 CLEAR and HOME</b>	22
2.1 The Screen - Rows and Columns	22
2.2 The Cursor	22
2.3 Scrolling	24
2.4 Cursor Control	24
2.5 Screen Design	22
2.6 Clearing the Screen and Homing the Cursor	26
2.7 Clearing the Screen and Homing the Cursor Using Escape Sequences	30
2.8 Setting Up Your Own Control String	32
2.9 A Screen-Based Program with ANSI Escape Sequence	32
2.10 Printing a Permanent Copy	33
2.11 Exercises	34

<b>3</b>	<b>Data Entry and Display</b>	35
3.1	Introduction	35
3.2	Simple Interest Example	35
3.3	Overall Design and Heading	36
3.4	Aligning the Prompts and Data Entry Fields	38
3.5	Spacing of Entries by Rows	38
3.6	Simple Interest Program - Heading, Prompts and Data Entry	39
3.7	Calculation and Display of Result	40
3.8	Extending the Program	42
3.9	Control Statements	42
3.10	Unconditional Control Statements	42
3.11	Conditional Control Statements	43
3.12	Simple Interest Program - Repeating the Run	44
3.13	Simple Interest Program - Amending the Data	44
3.14	Data Entry Screen Design	47
3.15	Exercises	51
<b>4</b>	<b>Errors and Error Messages</b>	53
4.1	Checking the Input	53
4.2	Generating an Error Message	53
4.3	Checking the Range	54
4.4	Checking the Data Type	59
4.5	Size of Data Fields - the LEN Function and VAL Function	61
4.6	Checking for Transcription Errors and Faulty Data	62
4.7	Sounding the Bell	63
4.8	Exercises	63
<b>5</b>	<b>Further Elements of BASIC</b>	65
5.1	Introduction	65
5.2	Arrays and Matrices	65
5.3	DIM Statement	65
5.4	FOR ... NEXT Loops	67
5.5	Subroutines	66
5.6	Special Matrix Commands	73
5.7	Random Numbers - RND and RANDOMIZE Functions	75
5.8	Rounding Numbers - The INT Function	76



5.9	Screen-based Example	77
5.10	Exercises	78
<b>6</b>	<b>Cursor Addressing</b>	<b>82</b>
6.1	Introduction	82
6.2	Moving the Cursor to a Specified Position	82
6.3	Moving the Cursor to a Specified Position Using ANSI Escape Sequences	84
6.4	Moving the Cursor to Point X, Y	85
6.5	Using Subroutines to Control the Cursor	90
6.6	A Screen-based Program with Cursor Addressing	91
6.7	Data Entry Screen Design with Cursor Addressing	93
6.8	Error Messages	100
6.9	Conclusion	100
6.10	Exercises	101
<b>7</b>	<b>Menu and Screen Systems</b>	<b>102</b>
7.1	Introduction	102
7.2	Menus	102
7.3	Loan Repayment Example	104
7.4	Designing the Screen System	105
7.5	Screen Numbering	108
7.6	Output on the Screen - Controlled Scrolling	109
7.7	Exercises	115
<b>8</b>	<b>Storing and Retrieving Data</b>	<b>116</b>
8.1	Introduction	116
8.2	READ, DATA and RESTORE Statements	116
8.3	Computer Files	117
8.4	Sequential Files	118
8.5	Adding Data to a Sequential File	122
8.6	Direct Access Files	122
8.7	Creating a Direct Access File	123
8.8	Reading Direct Access Records	125
8.9	Updating and Deleting Direct Access Records	125
8.10	Indexed Files	127
8.11	Creating an Indexed File	127
8.12	Reading Records in an Indexed File	129
8.13	Updating and Deleting Indexed Records	132

8.14	Reading an Indexed File in Alphabetical Order	133
8.15	Duplicate Names	134
8.16	Testing for the End of the File	134
8.17	Sorting with Indexed Files	135
8.18	Screen Systems Design for Data File Maintenance	136
8.19	Exercises	137
<b>9</b>	<b>Printed Reports</b>	141
9.1	Introduction	141
9.2	Directing Answers to a Printer	141
9.3	Headings and Titles	142
9.4	Printing Tables - PRINT USING	144
9.5	Counting, Totals and Averages	148
9.6	Exercises	151
<b>10</b>	<b>Program Design</b>	152
10.1	The Importance of Well-designed Programs	152
10.2	Structured Programs	153
10.3	Top Down Development	154
10.4	Developing Your Program	156
10.5	Types of Instruction	157
10.6	Documenting Your Program	158
<b>11</b>	<b>Introduction to Graphics</b>	161
11.1	Introduction	161
11.2	Graphics Characters	161
11.3	Histograms	163
11.4	Movement Effects	164
11.5	Reverse Video Effect	167
11.6	Exercises	169
<b>Appendix I</b>	<b>Summary of BASIC Commands</b>	170
<b>Appendix II</b>	<b>BASIC Functions</b>	180
<b>Appendix III</b>	<b>ASCII Characters</b>	188
<b>Appendix IV</b>	<b>Solutions to Exercises</b>	192
<b>Index</b>		232

# 1

# Elements of BASIC

## 1.1 What is a Computer?

A computer is a general purpose machine which can accept and store data in a form which represents letters, numbers and other punctuation and control characters. The computer can reorganize this data in various ways, including carrying out arithmetic, and can display the results obtained.

A computer cannot think or make decisions except in the purely mechanical sense of an automatic gearbox (which will change gear under certain preset physical conditions). Unlike a gearbox however a computer is not designed to carry out a particular task. Instead, it is designed to follow any set of instructions supplied to it in an appropriate form. A set of instructions is called a program.

## 1.2 What is a Program?

A computer program is a list of instructions which the computer carries out in order to complete a task. Each instruction is normally referred to as a statement. A program consists of a sequence of statements which the computer obeys or executes one at a time. The statements in a program are written in a carefully defined language consisting of certain words and symbols. Many different computer languages have been developed but only a few are widely used. Most languages consist of a precise and

logical form of English, together with mathematical symbols and functions.

BASIC is a programming language which is widely available, simple to learn and suitable for a wide range of applications.

The action of the computer in carrying out the instructions contained within the program is called processing. The end result of the processing will be to perform the task for which the program has been designed.

### **1.3 Components of a Computer**

The computer consists of the following physical components.

1. A keyboard, similar to a typewriter keyboard, for entering data.
2. A screen, similar to a television screen, for displaying programs, data and results.
3. A processing unit which contains electronic components which carry out or execute the program steps. The processing unit contains the work space in which the user's program is held while it is being worked on. This work space takes the form of random access memory (RAM).
4. A disk or tape unit which provides permanent storage for programs and data in computer readable form.
5. A printer for making a permanent copy of programs, data or results.

The keyboard and screen together are referred to as a visual display terminal. In a microcomputer, all of the above components will normally be grouped together in a single unit. In larger computers, several users can be supported simultaneously, and several separate terminals will be connected by cables to the processing unit.

## 1.4 Creating, Storing and Running a Program

To create a computer program, the user first switches the terminal on and then logs on using a pre-assigned user number and password. To enter the BASIC environment on the VAX system the user enters the word BASIC. The user can subsequently leave the BASIC environment by entering the word EXIT and can log off using LOGOFF. The computer will prompt the user to enter a command by displaying the word Ready. The procedure for logging on and off is illustrated in Figure 1-1.

This book is concerned with the details of the BASIC programming language. Initially, however, we need some general information about the commands used to control our programs. Suppose that we enter the following simple program

```
100 PRINT "HELLO"  
200 PRINT "MY NAME IS JOE"
```

Note that we enter a number at the start of each line. To go onto the next line we press the RETURN key. In our simple program we have numbered the lines 100 and 200. We could have used the numbers 1 and 2, but it often turns out to be useful to have gaps in the line numbering so that additional lines can be inserted. If we make a mistake when typing in a line of a program we can delete the incorrect characters by pressing a key marked DELETE or RUBOUT (or something similar depending on the terminal) and then retyping the characters correctly. The reader is advised against using the BACKSPACE key, as this does not delete the preceding character, even though that character will disappear from the screen when the next character is typed. The computer does not normally read the line of program or data until we press the RETURN key.

If we have already entered a line of program and we then decide that it is incorrect we can replace it by retyping the line correctly, starting with the original line number. We can delete a line entirely by typing just the line number and pressing the RETURN key. Additional lines

SCREEN FORMATTING SHEET

User	Application	Date
1		80
2		79
3		78
4		77
5		76
6		75
7		74
8		73
9		72
10		71
11		70
12		69
13		68
14		67
15		66
16		65
17		64
18		63
19		62
20		61
21		60
22		59
23		58
24		57
25		56
26		55
27		54
28		53
29		52
30		51
31		50
32		49
33		48
34		47
35		46
36		45
37		44
38		43
39		42
40		41
41		40
42		39
43		38
44		37
45		36
46		35
47		34
48		33
49		32
50		31
51		30
52		29
53		28
54		27
55		26
56		25
57		24
58		23
59		22
60		21
61		20
62		19
63		18
64		17
65		16
66		15
67		14
68		13
69		12
70		11
71		10
72		9
73		8
74		7
75		6
76		5
77		4
78		3
79		2
80		1
1	Username: JIMSMITH	
2	Password:	
3	Logged onto --TTJO: at 25-AUG-1983 10:12:27.87	
4	⚡ BASIC	
5		
6	VAX-11 BASIC V.-	
7		
8	Ready	
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20	Ready	
21		
22	EXIT	
23	⚡ LO	
24	JIMSMITH logged out at 25-AUG-1983 10:20:39.85	
81		80
82		79
83		78
84		77
85		76
86		75
87		74
88		73
89		72
90		71
91		70
92		69
93		68
94		67
95		66
96		65
97		64
98		63
99		62
100		61
101		60
102		59
103		58
104		57
105		56
106		55
107		54
108		53
109		52
110		51
111		50
112		49
113		48
114		47
115		46
116		45
117		44
118		43
119		42
120		41
121		40
122		39
123		38
124		37
125		36
126		35
127		34
128		33
129		32
130		31
131		30
132		29
133		28
134		27
135		26
136		25
137		24
138		23
139		22
140		21
141		20
142		19
143		18
144		17
145		16
146		15
147		14
148		13
149		12
150		11
151		10
152		9
153		8
154		7
155		6
156		5
157		4
158		3
159		2
160		1

Figure 1-1 Logging on and off

can be entered in any order and the computer will automatically insert them into the program in line number sequence.

The current status of the program can be displayed by entering the command

## LIST

This causes the computer to display the program in line number sequence.

To make the computer execute the program which is currently in the work space we enter the command

## RUN

In the present case the execution of the program will make the computer display the following

```
HELLO  
MY NAME IS JOE
```

If we log off, the program which is currently in the work space will be lost. In order to save the program for later use, we must give it a name, for example PROGA, and SAVE it. This is done by entering the command

## SAVE PROGA

This command causes the computer to make a copy of the program which is currently in the work space and store it on disk (or other medium) with the name PROGA. If an old version of program PROGA was already in storage it may be erased by this command. Subsequently we may retrieve the program by using the command

## OLD PROGA

This command will cause the computer to retrieve the program PROGA from storage on the disk and enter it into the work space. If another program is in the work space it will be erased. We can also erase the program which is currently in the work space by means of the command SCRATCH or NEW.

We can erase the permanent copy of PROGA from the disk by using the command

## **UNSAVE PROGA**

The VAX saves a BASIC program called PROGA, say, in the form of a file called PROGA.BAS;\*, where the asterisk (\*) represents a version number. The user can obtain a hard copy printout of this program by EXITing from BASIC and using the command

## **PRINT PROGA.BAS**

Several other useful commands for controlling programs are normally available, but the ones indicated so far are sufficient to enable the user to start using BASIC.

## **1.5 Errors and Debugging**

If your program contains an error it will not run correctly. In the development of a computer program it is normal for a number of errors to occur before the program runs successfully. The process of eliminating errors is called debugging.

If you make an error in entering a program statement, such that the computer cannot understand the statement, it is called a syntax error. When the computer encounters a syntax error it will stop and display an error message. The user then examines the program, corrects the error and tries again.

Sometimes a program will be free of syntax errors but will still not execute correctly. An execution error would



occur, for example, if we tried to make the computer divide by zero or read a number from a file which did not exist.

Finally, our program may contain logic errors. As far as the computer is concerned, these are not errors at all, since the machine is following the given instructions correctly. However, the program may not give the correct answer to the problem which the user had intended to solve. In this case the user has written the program incorrectly and will need to check carefully through it to find the mistake.

## 1.6 The PRINT Statement and the END Statement

The simplest BASIC program is one which will display a message on the screen. This involves the use of the command PRINT. The following is an example:

```
100 PRINT "HELLO"
```

The statement starts with a line number, in this case 100. The BASIC command PRINT is then given, which will cause the computer to print or display something on the screen. Then there is the text we wish to display which must be in quotation marks.

If we enter our simple one-line program into the computer and RUN it by typing the word RUN, the computer will display the word

```
HELLO
```

In some versions of BASIC it is necessary to conclude the program with an END statement. In this case the program to print HELLO would be as follows

```
100 PRINT "HELLO"  
200 END
```