MASTERING

COMPUTER PROGRAMMING

MACMILLAN MASTER SERIES

Accounting
Arabic
Astronomy
Australian History
Background to Business
Basic English Law
Basic Management
Biology
British Politics
Business Communication
Business Law
Business Microcomputing
Catering Science
Catering Theory
Chemistry
COBOL Programming
Commerce
Computer Programming
Computers
Data Processing
Economic and Social History
Economics
Electrical Engineering
Electronics
English as a Foreign Language
English Grammar
English Language
English Literature
Financial Accounting
French
French 2

German
German 2
Hairdressing
Italian
Italian 2
Japanese
Keyboarding
Marketing
Mathematics
Modern British History
Modern European History
Modern World History
Nutrition
Office Practice
Pascal Programming
Physics
Practical Writing
Principles of Accounts
Restaurant Service
Social Welfare
Sociology
Spanish
Spanish 2
Spreadsheets
Statistics
Statistics with your
Microcomputer
Study Skills
Typewriting Skills
Word Processing

MASTERING COMPUTER PROGRAMMING

THIRD EDITION

P. E. GOSLING

M
MACMILLAN

© Peter Gosling 1982, 1984, 1990

All rights reserved. No reproduction, copy or transmission of this publication may be made without written permission.

No paragraph of this publication may be reproduced, copied or transmitted save with written permission or in accordance with the provisions of the Copyright Act 1956 (as amended), or under the terms of any licence permitting limited copying issued by the Copyright Licensing Agency, 33-4 Alfred Place, London WC1E 7DP.

Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

First edition 1982

Reprinted 1982, 1983

Second edition 1984

Reprinted 1985

Third edition 1990

Published by

MACMILLAN EDUCATION LTD

Houndmills, Basingstoke, Hampshire RG21 2XS

and London

Companies and representatives

throughout the world

British Library Cataloguing in Publication Data

Gosling, P. E. (Peter Edward)

Mastering computer programming.—(3rd ed.).

1. Computer systems. Programming languages: Basic language

I. Title II. Series

005.13'3

ISBN 978-0-333-51005-6

ISBN 978-1-349-11094-0 (eBook)

DOI 10.1007/978-1-349-11094-0

CONTENTS

<i>Preface</i>		viii
<i>Glossary of trademarks</i>		x
1 What is a computer program?	1.1 Introduction	1
	1.2 The user as programmer	2
	1.3 Planning your program	3
	1.4 Storing data in memory	6
	1.5 Disk storage	7
	1.6 Serial and random files	8
	1.7 Sorting	9
	1.8 Handling characters	9
2 BASIC: a popular language	2.1 Introduction	10
	2.2 BASIC	12
	2.3 Examples	21
	2.4 Compilers and Interpreters	24
3 Decisions, decisions	3.1 Conditional and unconditional jumps	25
	3.2 Putting decisions to use	27
	3.3 WHILE/WEND	29
	3.4 FOR/NEXT loops	33
4 Storing data in memory	4.1 Storing lists	37
	4.2 Sorting lists into order	39
	4.3 Tables	46
	4.4 Strings	48
5 The electronic filing cabinet	5.1 Introduction	51
	5.2 Serial files	52
	5.3 Use of files in programs	56
	5.4 Files in data processing	60
	5.5 Random access files	66

6	Subroutines	6.1	A musical digression	73
		6.2	Down to business	75
		6.3	Home-made functions	77
		6.4	BASIC subroutines	78
		6.5	Called subroutines	80
7	Graphics in BASIC	7.1	Introduction	85
		7.2	Line and box drawing	86
		7.3	Circles	90
		7.4	Painting	92
8	Putting programming to work	8.1	Introduction	94
		8.2	Getting started	95
		8.3	Creating another program	97
		8.4	Filling in the details	103
		8.5	Some variations	114
9	Bug hunting, or why my program never works first time	9.1	Introduction	116
		9.2	Typing errors	116
		9.3	Wrongly labelled instructions	117
		9.4	Wrongly nested loops	117
		9.5	Hard loops	118
		9.6	Problems with loops	119
		9.7	Interpreter errors	121
		9.8	Choosing test data	122
		9.9	Expecting the computer to do the impossible	123
10	The dBASE programming language	10.1	Introduction	125
		10.2	The electricity bill program	126
		10.3	A typical dBASE program	127
		10.4	dBASE menus	132
		10.5	Indexing files	135
		10.6	Finally	137
11	COBOL	11.1	Introduction	138
		11.2	COBOL program layout	140
		11.3	A COBOL program handling files	142
12	Pascal	12.1	Introduction	145
		12.2	Pascal instructions	146
		12.3	A binary search program	146
		12.4	The electricity bill program	149

	12.5	A Turbo Pascal program using files	150
13	C	13.1 Introduction	155
		13.2 A program in C	156
		13.3 The electricity bill again	158
		13.4 Loops in C	158
14	PROLOG	14.1 Introduction	161
		14.2 Logic systems	169
		14.3 Why use PROLOG?	172
		14.4 The PROLOG approach	173
		14.5 Using Turbo PROLOG	175
		14.6 Another example	179
		14.7 A word of warning	180
		<i>Further Reading</i>	184
		<i>Index</i>	186

PREFACE

Despite the fact that so many specialised “packages” have come onto the market since the first edition of this book was published in 1982 there is still a need for people to learn the art of programming. Many of these packages possess their own internal programming facility often cloaked under the name of a *macro*. Even the WordPerfect word processing program, on which this book was written, has a powerful programming language available. The macro facility in the Lotus 1–2–3 and SuperCalc spreadsheets enable you to do things with them that one would never have dreamed of in the days when all spreadsheets did was to update stock lists and perform complex calculations at great speed. You can see how these macros are created in my book *Mastering Spreadsheets*.

So this book is intended to be a guided tour of programming and what it involves. Its object, as with the previous two editions, is to introduce the techniques of writing computer programs. The reader should realise that the first step in writing a program is that it should be *planned* carefully. The most important part of programming is the *method* that is adopted to achieve the required end. The rest is then to decide which *language* is best suited to the job in hand and finally the end of the task is to *code* the program in that language. As you will see, you would not use a language such as PROLOG to perform mathematical calculations, nor would you use C for handling files of numerical data. “Horses for Courses” is the watchword in programming.

In the preparation of this book I must extend my grateful thanks to Lauretta Hooper of Keene Business to Business for supplying me with the latest versions of Borland International’s Turbo Pascal, Turbo C and Turbo PROLOG. My colleague Frank Hatfull has been of great service as well in offering help and suggestions together with a program that I would never have dreamed of writing myself.

I have tried to produce as many varied examples as possible; I hope that we have a sort of “Hitch-hiker’s Guide to Programming” so that the reader will gain the flavour of the various programming languages dealt with. This book in no way attempts to give detailed descriptions, which is why the reader is pointed in the direction of more comprehensive texts (see Further Reading) in order to gain an in-depth understanding of a particular language.

The important thing about programming is that it requires *practice*. It is very much a “doing” activity and reading all the books in the world will not make a competent programmer. You will only learn by your mistakes, as the author well knows in the preparation of the examples for this book. Remember the golden rules

NO PROGRAM OF OVER FOUR LINES EVER WORKS FIRST TIME

DON’T PANIC

DON’T GIVE UP

February 1989

Peter Gosling

GLOSSARY

OF TRADEMARKS

dBASE	Ashton-Tate Compnay
dBASE III PLUS	Ashton-Tate Company
GWBasic	MicroSoft Corporation
MS-DOS	MicroSoft Corporation
Turbo Pascal	Borland International
Turbo C	Borland International
Turbo PROLOG	Borland International