C Programming

Other titles of related interest

Ian O. Angell and Dimitrios Tsoubelis, *Advanced Graphics on VGA and XGA Cards Using Borland C++*

N. Frude, *A Guide to SPSS/PC+, second edition*

Peter Grossman, *Discrete Mathematics for Computing*

Percy Mett, *Introduction to Computing*

Tony Royce, *COBOL - An Introduction*

Tony Royce, *Structured COBOL - An Introduction*

C Programming

Tony Royce

palgrave
macmillan

© A. P. Royce 1996

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, Designs and Patents Act 1988, or under the terms of any licence permitting limited copying issued by the Copyright Licensing Agency, 90 Tottenham Court Road, London W1T 4LP.

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

The author has asserted his right to be identified as the author of this work in accordance with the Copyright, Designs and Patents Act 1988.

Published by

PALGRAVE MACMILLAN

Houndmills, Basingstoke, Hampshire RG21 6XS and

175 Fifth Avenue, New York, N. Y. 10010

Companies and representatives throughout the world

PALGRAVE MACMILLAN is the global academic imprint of the Palgrave Macmillan division of St. Martin's Press, LLC and of Palgrave Macmillan Ltd. Macmillan® is a registered trademark in the United States, United Kingdom and other countries. Palgrave is a registered trademark in the European Union and other countries.

ISBN 978-0-333-63851-4

ISBN 978-1-349-13759-6 (eBook)

DOI 10.1007/978-1-349-13759-6

This book is printed on paper suitable for recycling and made from fully managed and sustained forest sources.

A catalogue record for this book is available from the British Library.

12 11 10 9 8 7
08 07 06 05 04

Contents

Preface

viii

Section 1 - Introductory C Language Topics

1.1	Introduction to Programming	1
1.2	Programming in C	2
1.3	Structure of a C Program	3
1.4	Layout of a C Program	4
1.5	Vocabulary	5
1.6	Variables	6
1.7	Using scanf and printf	7
1.8	Strings	9
1.9	Assignment	10
1.10	Arithmetic	11
1.11	Arithmetic using Assignment Operators	12
1.12	Incrementing and Decrementing Variables	13
1.13	Using if .. else (1)	14
1.14	Using if .. else (2)	15
1.15	Using if .. else (3)	16
1.16	Using Logical Operators (&&, , !)	17
1.17	Using switch	18
1.18	Repetition using while	19
1.19	Repetition using do .. while	20
1.20	Repetition using for	21
1.21	Functions	22
1.22	Functions - Local Variables	24
1.23	Functions - Passing Parameters	26
1.24	Functions - Using return	28
1.25	Functions - Repetitive Calls	30
1.26	Functions - Passing Work on to Other Functions	32
1.27	Recursion	34
1.28	Global Variables	35
1.29	Static Local Variables	37
1.30	Using #define	38
1.31	Using enum	40
1.32	String Functions	42
1.33	Blocks	43
1.34	Using break/continue/exit/return/goto	44
1.35	Using printf - Formatting Output	46
1.36	Using scanf - Specifying Input Format	48
1.37	Using putchar/getchar	49
1.38	Buffered Keyboard Input	50
1.39	Evaluating a Condition	52
1.40	Data Types	56

Section 2 - Program Design, Implementation and Testing

2.1	Introduction to Program Design	59
2.2	Program Design - Sequences	60
2.3	Program Design - Iteration	61
2.4	Program Design - Selection	62
2.5	Program Design - Combining Structures	64
2.6	Implementing a Design in C	70
2.7	In-line Coding from a Structure Diagram - Sequences	71

2.8	In-line Coding from a Structure Diagram - Iteration	75
2.9	In-line Coding from a Structure Diagram - Selection	77
2.10	Implementing a Complex Program Design using In-line Code	79
2.11	Hierarchical Coding from a Structure Diagram - Sequences	83
2.12	Hierarchical Coding from a Structure Diagram - Iteration	86
2.13	Hierarchical Coding from a Structure Diagram - Selection	88
2.14	Implementing a Complex Program Design using Hierarchical Code	90
2.15	Implementing a Program Design using In-line and Hierarchical Code	93
2.16	Testing Software	96

Section 3 - Complex Data Structures

3.1	Defining Structures	101
3.2	Using Structures	102
3.3	Defining Structure Types	104
3.4	Using Structure Types	105
3.5	Operations on Structures	106
3.6	Defining Nested Structures	109
3.7	Using Nested Structures	111
3.8	Defining Arrays - Single Dimensional	113
3.9	Using Arrays - Single Dimensional	114
3.10	Defining Arrays - Two Dimensional	116
3.11	Using Arrays - Two Dimensional	117
3.12	Arrays of Strings	119
3.13	Defining Arrays of Structures	121
3.14	Using Arrays of Structures	122
3.15	Central Memory	124
3.16	Pointers	126
3.17	Using Pointers	128
3.18	Passing Pointers as Parameters	130
3.19	Pointers to Structures	131
3.20	Pointers and Arrays	133
3.21	Passing Arrays as Parameters	137
3.22	Arrays of Pointers	139
3.23	Pointers to Pointers	142
3.24	Pointers to Arrays of Pointers	143
3.25	Dynamic Memory Allocation - Requesting and Releasing Space	145
3.26	Dynamic Memory Allocation - Using Space	147
3.27	Linked Lists	149
3.28	Circular Linked Lists	158
3.29	Queues	167
3.30	Trees	173
3.31	Files	184
3.32	Files - Storing Data on Disk using fwrite	185
3.33	Files - Reading Data from Disk using fread	188
3.34	Files - Writing Formatted Text to Disk using fprintf	191
3.35	Files - Reading Formatted Text from Disk using fscanf	193
3.36	Files - Writing Data to Disk - one byte at a time - using putc	196
3.37	Files - Reading Data from Disk one byte at a time using getc	198
3.38	Random Files - using fseek	200
3.39	Files - Handling Errors	205

Section 4 - Modular Programming

4.1	Introduction to Modular Programming	212
4.2	Using Functions from other Modules	213
4.3	Using Variables from other Modules	216
4.4	Static Global Variables	218
4.5	Object Libraries	221

Section 5 - Advanced and Supplementary C Language Topics

5.1	Macros	224
5.2	Macros - using Arguments	226
5.3	Using typedef	228
5.4	Conditional Expressions	230
5.5	Defining Unions	231
5.6	Using Unions	232
5.7	Unions and Structures	233
5.8	Register Variables	236
5.9	Using const and volatile	237
5.10	Using Bitwise Operators (&, , ^, >>, <<)	239
5.11	Defining Bit-fields	245
5.12	Using Bit-fields	247
5.13	Standard Character Type Functions	248
5.14	Using atoi, atof, atol	250
5.15	Pointers to Functions	251
5.16	Arrays of Pointers to Functions	255
5.17	Passing Function Addresses as Parameters	257
5.18	Command Line Arguments	259
5.19	Conversion of Data Types	261
5.20	Using a Cast	264

Appendices

A - Developing a C Program on the UNIX system	265
B - Using vi	266
C - The Binary System	267
D - The Hexadecimal System	270
E - Character Codes	271

<i>Index</i>	273
--------------	-----

Preface

This text has been written for students with little or no previous programming experience. It should be useful on a wide range of courses - in particular - degree, BTEC, City and Guilds and various professional courses in the C language.

The text substantially covers the C language (ANSI standard) and introduces program design and testing.

As far as possible, each lesson is short and covers a single topic. There are numerous examples and exercises. Topics have been arranged in order of difficulty and usefulness. Easy, and more commonly used, subjects are at the beginning and harder, or less used, topics are near the end.

The lessons are grouped into five sections:

Introductory C Language Topics

Program Design, Implementation and Testing

Complex Data Structures (Structures, Arrays, Linked Lists, Queues, Trees, Files)

Modular Programming

Advanced and Supplementary C Language Topics

It is intended that the reader should start practical programming exercises as soon as possible and should be able to start writing simple programs from Page 1.

It is important to try out all the practical examples on a computer and to attempt all the exercises. Just as with any language, programming languages become easy with constant use.

I should like to thank students and colleagues for their help and encouragement in the development of this book.