Mastering

C Programming

Palgrave Master Series

Accounting	Global Information Systems
Accounting Skills	Human Resource Management
Advanced English Language	Information Technology
Advanced English Literature	International Trade
Advanced Pure Mathematics	Internet
Arabic	Italian
Basic Management	Java
Biology	Management Skills
British Politics	Marketing Management
Business Communication	Mathematics
Business Environment	Microsoft Office
C Programming	Microsoft Windows, Novell
C++ Programming	NetWare and UNIX
Chemistry	Modern British History
COBOL Programming	Modern European History
Communication	Modern United States History
Computing	Modern World History
Counselling Skills	Networks
Counselling Theory	Novels of Jane Austen
Customer Relations	Organisational Behaviour
Database Design	Pascal and Delphi Programming
Delphi Programming	Philosophy
Desktop Publishing	Physics
Economic and Social History	Practical Criticism
Economics	Psychology
Electrical Engineering	Shakespeare
Electronics	Social Welfare
Employee Development	Sociology
English Grammar	Spanish
English Language	Statistics
English Literature	Strategic Management
Fashion Buying and Merchandising	Systems Analysis and Design
Management	Team Leadership
Fashion Styling	Theology
French	Twentieth Century Russian History
Geography	Visual Basic
German	World Religions

www.palgravemasterseries.com

Palgrave Master Series

Series Standing Order ISBN 0-333-69343-4

(outside North America only)

You can receive future titles in this series as they are published by placing a standing order. Please contact your bookseller or, in case of difficulty, write to us at the address below with your name and address, the title of the series and the ISBN quoted above.

Customer Services Department, Macmillan Distribution Ltd
Houndmills, Basingstoke, Hampshire RG21 6XS, England

Mastering

C Programming

W. Arthur Chapman

© W. Arthur Chapman 1991

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 978-0-333-49842-2

ISBN 978-1-349-11015-5 (eBook)

DOI 10.1007/978-1-349-11015-5

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.

13 12 11 10 9 8
07 06 05 04 03 02

Contents

<i>List of Figures and Tables</i>	ix
<i>Preface</i>	xi
1. Beginning with problems	
1.1 Preliminaries	1
1.2 Problem solving	2
1.3 Devising a solution	4
1.4 Algorithm	8
1.5 Programming	11
1.6 Pseudocode	14
1.7 A line editor	16
1.8 The computer program	21
Summary	24
Exercises	25
2. Towards C	
2.1 Introduction	27
2.2 The first C program	27
2.3 C program structure	31
2.4 Functions – a first look	33
2.5 From code to results	37
Summary	40
Exercises	41
3. Of words and objects	
3.1 Language	43
3.2 Data types	48
3.3 Making declarations	54
3.4 Doing a little calculating	58
3.5 Operators	63
3.6 Some new operators	69
3.7 Type conversion	70
3.8 Expressions	73
3.9 Statements	74
3.10 Formatted input and output: <i>scanf()</i> and <i>print()</i>	76
Summary	81
Exercises	81

4. Selection in C, or ‘Which way next?’	
4.1 Introduction	84
4.2 Conditional expressions	84
4.3 Logical operators	86
4.4 The <i>IF</i> statement	89
4.5 The <i>IF ... ELSE</i> statement	93
4.6 The <i>SWITCH</i> statement	96
Summary	100
Exercises	101
5. Doing it again and again!	
5.1 Introduction	103
5.2 The <i>WHILE</i> loop	103
5.3 The <i>DO ... WHILE</i> loop	109
5.4 The <i>FOR</i> loop	111
5.5 The comma operator	115
5.6 Example – prime numbers	118
5.7 Arrays – a quick look	119
Summary	121
Exercises	122
6. Functions – making them useful	
6.1 Introduction	124
6.2 The <i>RETURN</i> statement	124
6.3 Function types other than <i>int</i>	129
6.4 Passing data into a function	132
6.5 Some examples	135
6.6 Call by value	140
6.7 Functions and header files	144
6.8 Storage class	146
Summary	150
Exercises	150
7. The calculator	
7.1 Introduction	153
7.2 Problem definition	153
Exercises	165
8. Pointers, arrays and strings	
8.1 Introduction	166
8.2 Pointers	166
8.3 Pointers and functions – call by reference	169
8.4 Arrays	172
8.5 Arrays and pointers	175
8.6 Arrays and functions	179
8.7 Strings	187

8.8	String library functions	193
8.9	Arrays of strings	196
8.10	Two-dimensional arrays	197
8.11	More on pointers	198
8.12	Using arrays – the bridge tutor	201
	Summary	203
	Exercises	204
9.	Input and output – more thoughts	
9.1	Introduction	206
9.2	Input and output – the story so far	206
9.3	More on formatting using <i>printf()</i>	208
9.4	Input formatting using <i>scanf()</i>	211
9.5	File i/o	213
9.6	Doing some file i/o	216
	Summary	226
	Exercises	227
10.	<i>Typedef</i>, structures and unions	
10.1	Introduction	228
10.2	<i>TYPEDEF</i>	228
10.3	Structures	232
10.4	Structures and <i>TYPEDEF</i>	236
10.5	Structures within structures	238
10.6	Pointers and structures	241
10.7	Unions	244
	Summary	247
	Exercises	247
11.	Lists and list processing	
11.1	Introduction	251
11.2	Basic concepts	251
11.3	Simple lists	259
11.4	More list processing functions	260
	Summary	270
	Conclusion	270
	<i>Appendix A</i>	
	The ASCII codes	271
	<i>Appendix B</i>	
	The line editor	272
	<i>Appendix C</i>	
	The bridge tutor	288

viii *Contents*

<i>Appendix D</i>	
Further reading	302
<i>Index</i>	303

List of Figures and Tables

FIGURES

1.1	Problem solving ... 'How do I cross the road?'	2
1.2	The desk instructions	7
1.3	Simple program control	12
1.4	Program control with selection	13
1.5	The line editor	19
1.6	The life cycle of a computer program	23
2.1	Simple C program structure	32
2.2	Bridge Tutor main-menu screen	37
2.3	From code to results - the likely route in C	39
3.1	From characters to a program	45
3.2	Bits, bytes and words	49
3.3	Float numbers	52
3.4	The apothecary's window	58
3.5	Incrementing	64
3.6	Using brackets to change the order of evaluation	67
5.1	The structure of a <i>for</i> loop	112
7.1	The <i>process()</i> algorithm	158
8.1	Using a pointer	168
8.2	Array representation	173
8.3	Using a pointer to index an array	176
8.4	Comparison of a single character stored in an array (<i>*char</i>) and stored as a <i>char</i>	187
8.5	Arrays of strings	197
11.1	A list element	252
11.2	Addressing the first element of a list	252
11.3	A simple list	253
11.4	Converting a two-character string to a list using recursion	255
11.5	Inserting elements into a list	263
11.6	Insertion in the middle of a list	264
11.7	Deleting an element from a list	266
11.8	Deleting by copying the next element	266
11.9	A list requiring ordering	268
11.10	The ordered list	268

TABLES

1.1	Valid line definitions for the line editor	18
1.2	The line editor commands	20
3.1	The keywords of C	48
3.2	Associativity and precedence of the arithmetic operators	66
3.3	The arithmetic assignment operators	70
3.4	White spaces - nonprinting characters	77
3.5	Common conversion specifications	78
4.1	The relational operators	85
4.2	The truth-table for <code>&&</code> and <code> </code> in terms of truth values	87
4.3	The truth-table for <code>&&</code> and <code> </code> for any expression	88
4.4	Truth-table for the expression: $(x \neq 0 \ \&\& \ 1/x > 0.001)$	89
7.1	Evaluation of the expression: $2*(4+3*(7-5))/(10-6)$	159
8.1	The relationship between array addresses, array elements, array contents and pointers	180
9.1	Standard input/output functions	207
9.2	<i>printf()</i> conversion-string examples: <i>char</i> and <i>int</i>	209
9.3	<i>printf()</i> conversion characters	210
9.4	<i>printf()</i> conversion-string examples: <i>float</i>	211
9.5	<i>printf()</i> conversion-string examples: strings	211
9.6	<i>scanf()</i> conversions (main set)	212
9.7	Input/output mode parameters	214
9.8	File i/o functions and their <i>stdio</i> equivalents	216

Preface

This book is intended as a first course in C programming. It is suitable for those new to programming as well as for those already familiar with another programming language. Access to a computer running C is assumed. With this condition the text is suitable for use in self-study, directed study through open or distance learning as well as via a more traditional approach as a class text. All the example programs and functions have been tested using Turbo C version 2.0. However, with very few exceptions, no changes should be necessary if other C compilers are used.

The main aim of the book is to introduce C and to provide the essentials of the language. The standard used throughout is the draft ANSI standard, and its counterpart the draft British Standard Specification (ISO/IEC DIS 9899), which is summarised in the second edition of the classic text for C *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie, Englewood Cliffs, NJ: Prentice Hall 1988.

The text begins by looking at problem solving in fairly general terms before moving on to a first C program in Chapter 2. These first two chapters form an important introduction to the main text and are especially intended for anyone new to programming. Subsequent chapters develop the C language, its syntax and semantics. The material is designed to take the reader step by step from the basics (Ch. 3) through control structures (Chs. 4–6) and data structures (Chs. 8 and 10) to the more advanced topics of lists and list processing (Ch. 11). New elements of the C language are illustrated by numerous examples of program fragments, functions and complete programs.

Throughout the book a number of rather more substantial programs are developed to provide a context for the use of C in rather larger projects. As and when appropriate, these programs, their design and implementation are discussed and functions developed. Three main programs are dealt with in detail; they are a calculator, a line editor and a simple bridge tutor. The calculator is discussed in detail, and the program is developed, in Chapter 7. We introduce the line editor in Chapter 1 and develop various parts of it throughout the book. The bridge tutor is introduced in Chapter 2 and provides a simple program to simulate shuffling, dealing, counting

points and suggesting opening bids. For the most part no knowledge of bridge is necessary but a familiarity with cards and card games such as whist would be helpful. These latter two programs are listed in full, and their functions are discussed in detail, in Appendices B and C respectively.

Most chapters conclude with a summary which highlights the main points covered in the chapter and which serves to act as a revision aid to the reader. In addition, most chapters contain exercises which are designed to reinforce the topics covered and to develop the readers understanding of C. Some of these exercises refer to the larger programs and as such the answers can be found in the relevant program listings.

As you work through the material presented here you should develop a good understanding of C and C programming. If by the time you have completed your study of this text you have a desire to continue programming in C, wish to move on to more advanced aspects of the language, and have even more importantly found that C programming is both challenging and also fun, then the book will have achieved its purpose.

This book developed out of an idea suggested by my friend and colleague Noel Chidwick and I would like to thank him for that original idea and his encouragement throughout the project. (Not to mention the late nights and early mornings which seemed to form an inevitable part of life in recent months!) Thanks are also due to many other friends and colleagues at Telford College, Edinburgh and further afield who have helped and supported me in various ways. I would also like to extend my gratitude to students who attended various classes given by me in recent years. They willingly tried out many of the ideas which finally found their way into this book and provided much helpful stimulation. I am pleased to be able to extend my thanks to Jane Wightwick at the publishers for her help and support during the lifetime of this project and her understanding when deadlines were missed. Finally, and most importantly, I would like to thank my wife Judy and our children Emma, Lucy and Donald who have put up with an, even more than usual, bad-tempered fifth member of the household! Without their forbearance and encouragement the task of writing would have been much harder.