Macmillan Computer Science Series

Consulting Editor:

Professor F.H. Sumner, University of Manchester

- A. Abdellatif, J. Le Bihan, M. Limame, *Oracle - A User's Guide*
Ian O. Angell, *High-resolution Computer Graphics Using C*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using FORTRAN 77*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using Pascal*
M. Azmoodeh, *Abstract Data Types and Algorithms, second edition*
C. Bamford and P. Curran, *Data Structures, Files and Databases, second edition*
P. Beynon-Davies, *Information Systems Development, second edition*
G.M. Birtwistle, *Discrete Event Modelling on Simula*
Richard Bornat, *Understanding and Writing Compilers*
Linda E.M. Brackenbury, *Design of VLSI Systems - A Practical Introduction*
Alan Bradley, *Peripherals for Computer Systems*
G.R. Brookes and A.J. Stewart, *Introduction to occam2 on the Transputer*
P.C. Capon and P.J. Jinks, *Compiler Engineering Using Pascal*
Robert Cole, *Computer Communications, second edition*
Eric Davalo and Patrick Naïm, *Neural Networks*
S.M. Deen, *Principles and Practice of Database Systems*
C. Delannoy, *Turbo Pascal Programming*
Tim Denvir, *Introduction to Discrete Mathematics for Software Engineering*
D. England *et al.*, *A Sun User's Guide, second edition*
Jean Ettinger, *Programming in C++*
J.S. Florentin, *Microprogrammed Systems Design*
A.B. Fontaine and F. Barrand, *80286 and 90386 Microprocessors*
Michel Gauthier, *Ada - A Professional Course*
M.G. Hartley, M. Healey and P.G. Depledge, *Mini and Microcomputer Systems*
J.A. Hewitt and R.J. Frank, *Software Engineering in Modula-2 - An Object-oriented Approach*
Roger Hutty, *COBOL 85 Programming*
Patrick Jaulent, *The 68000 - Hardware and Software*
M.J. King and J.P. Pardoe, *Program Design Using JSP - A Practical Introduction, second edition*
Bernard Leguy, *Ada - A Programmer's Introduction*
M. Léonard, *Database Design Theory*
David Lightfoot, *Formal Specification Using Z*
A.M. Lister and R.D. Eager, *Fundamentals of Operating Systems, fifth edition*
Elizabeth Lynch, *Understanding SQL*
Tom Manns and Michael Coleman, *Software Quality Assurance, second edition*
B.A.E. Meekings, T.P. Kudrycki and M.D. Soren, *A book on C, third edition*
R.J. Mitchell, *C++ Object-oriented Programming*
R.J. Mitchell, *Microcomputer Systems Using the STE Bus*
R.J. Mitchell, *Modula-2 Applied*
Y. Nishinuma and R. Espesser, *UNIX - First Contact*
Ian Pratt, *Artificial Intelligence*
Pham Thu Quang and C. Chartier-Kastler, *MERISE in Practice*
A.J. Pilavakis, *UNIX Workshop*

continued overleaf

E.J. Redfern, *Introduction to Pascal for Computational Mathematics*
F.D. Rolland, *Programuning with VDM*
A.G. Sutcliffe, *Human-Computer Interface Design*
C.J. Theaker and G.R. Brookes, *Concepts of Operating Systems*
M. Thorin, *Real-time Transaction Processing*
M.R. Tolhurst *et al.*, *Open Systems Interconnection*
A.J. Tyrell, *COBOL from Pascal*
I.R. Wilson and A.M. Addyman, *A Practical Introduction to Pascal, second edition*

Other titles

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*
Percy Mett, *Introduction to Computing*
Tony Royce, *COBOL - An Introduction*
Tony Royce, *Structured COBOL - An Introduction*

Programming in C++

Jean Ettinger MSc CEng

*School of Computer Science and Information Systems Engineering
University of Westminster*

M

MACMILLAN

© Jean Ettinger 1994

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 provision 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 W1P 9HE.

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

First published 1994 by
THE MACMILLAN PRESS LTD
Houndmills, Basingstoke, Hampshire RG21 2XS
and London
Companies and representatives
throughout the world

ISBN 978-0-333-60682-7   ISBN 978-1-349-23304-5 (eBook)
DOI 10.1007/978-1-349-23304-5

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

A 3.5 in. MS-DOS diskette containing all the listings of the program examples is available at the price of £25.00 or \$40.00 (inc. postage/ packing - airmail overseas).

Please send your personal cheque to: Globe Book Services Ltd,
Houndmills, Basingstoke, Hampshire, RG21 2XS, UK.

Access/ Visa/ American Express/ Diners Club accepted.

Please quote number and expiry date.

Please quote ISBN 978-0-333-62295-7 when ordering.

Turbo C®++ for Windows is copyrighted ©1987-1992, by Borland International Inc., all rights reserved.

Diagrams and text from Turbo C++ for Windows are reprinted with permission.

Turbo C is a registered trademark of Borland International Inc.

*To the memory of my husband, Charles,
who always supported my career beyond duty,
and to our excellent teenagers, Andrew and Beth*

Contents

<i>Preface</i>	xi
1 The inheritance of C++	1
1.1 The Algol family of computer languages	1
1.2 Smalltalk, an object-oriented language	4
1.3 An overview of C++	5
Exercises for Chapter 1	7
2 Simple programs, and the predefined data types	8
2.1 The first program	8
2.2 Introduction to input and output using the iostream library	9
2.3 Identifiers	11
2.4 The predefined data types	11
2.5 Declarations, initialisations, constants	11
2.6 The predefined integer types	12
2.7 The predefined types for fractional numbers	14
2.8 The assignment, compound assignment, and sizeof operators	15
2.9 Type conversions	17
2.10 Data objects	18
2.11 Using libraries	18
Exercises for Chapter 2	19
3 Structured programming	21
3.1 Boolean values, boolean expressions	21
3.2 The selection statements: if, if else, and switch	23
3.3 The increment and decrement operators	25
3.4 The repetition statements: while, do, and for	26
3.5 The break and continue statements	28
Exercises for Chapter 3	29
4 Derived types	30
4.1 The pointer type	30
4.2 Reference types	32
4.3 Arrays	33
4.4 Strings	37
4.5 Using the string library	38
4.6 Enumerated constants	41
4.7 Structs and the selection operators	41
Exercises for Chapter 4	44

5 Functions	45
5.1 Function definition, declaration and call	45
5.2 Passing arguments by value and by reference	48
5.3 Return values and the return statement	50
5.4 Inline functions	50
5.5 The comma operator	51
5.6 The first version of a program that plays the hangman game	51
5.7 The scope, visibility, and duration of an identifier	58
5.8 The conditional operator ?:	60
5.9 Program arguments	60
Exercises for Chapter 5	61
6 Data files and library files	62
6.1 Handling data file streams	62
6.2 The second version of the hangman game using a data file	64
6.3 Creating user-defined libraries	65
6.4 The third version of the hangman game with a user library	66
6.5 Encapsulation and implementation hiding	69
Exercises for Chapter 6	71
7 Classes and derived classes	72
7.1 Defining a class	72
7.2 Handling class objects	74
7.3 The fourth version of the hangman game showing the use of a class	75
7.4 Inheritance from a base class by a derived class	78
7.5 The fifth version of the hangman game showing class inheritance	81
Exercises for Chapter 7	84
8 Overloading and friends	85
8.1 Program example with overloading and friend functions	85
8.2 Friends	89
8.3 Function overloading	90
8.4 Operator overloading	91
8.5 Operator symbol definitions	95
8.6 Classes are user-defined types	95
Exercises for Chapter 8	96
9 Dynamic memory allocation	97
9.1 Simply linked lists	98
9.2 Defining base classes for a simply linked list	99
9.3 Implementing an actual linked list structure	102
9.4 Allocating and deallocating free store: the new and delete operators	102

9.5 Defining a class with a constructor using the free store	104
9.6 Explicitly converting pointer referenced types	105
9.7 The library files defining an actual simply linked structure	106
9.8 A program example using the actual linked list definition	107
Exercises for Chapter 9	110
10 Templates	111
10.1 The template formal type parameter list	111
10.2 Function templates	112
10.3 Constructor initialisation lists	115
10.4 Class templates with generic parameters	116
10.5 Class templates using constant expression parameters	118
10.6 Reservations on the current templates definition	120
Exercises for Chapter 10	121
11 Virtual functions and virtual base classes	122
11.1 Selecting virtual functions	122
11.2 Example program illustrating virtual functions	122
11.3 Pure virtual functions and abstract base classes	127
11.4 Virtual base classes	127
11.5 Example program handling a hierarchy of classes	127
11.6 Conclusion	135
Exercises for Chapter 11	135
Appendix 1 Using Unix GNU C++ and Borland Turbo C++	137
Appendix 2 Escape sequences and ASCII charts	140
Appendix 3 C++ keywords	141
Appendix 4 Operator precedence and associativity	142
Appendix 5 The iostream libraries	143
Appendix 6 Other libraries	149
Index	153

Preface

This book has been written to meet the demand for an introductory text explaining the features of the C++ language.

The recent widespread availability of C++ implementations for PCs and Unix systems has encouraged both individuals and businesses to explore its use. Many people coming to these implementations have had no previous programming experience, or have a background in computer languages that have little in common with the structured approach of those in the C++ family of languages.

Moreover, there is a trend to use C++ as a first language for students on computing courses, since it provides a vehicle to introduce many of the current software engineering concepts which aid the production of robust and reusable software.

During the past year, many helpful comments and suggestions for improvements of the material for this text have been made by students studying for BSc computing Degrees, students on an MSc Software Engineering course, and students on several C++ evening Short Courses. A number of these students have been seconded from industry, where many software teams are currently converting to the use of C++ implementations.

There is, as yet, no ANSI standard for C++. AT&T Bell Laboratories are introducing successive versions onto the marketplace. This book discusses C++ in terms of the AT&T version 2.1.

The book assumes no prior knowledge of the earlier C language. All the main features of C++ are discussed, from the beginning, with many program examples to illustrate them. Exercises are provided, and all of the programs that they suggest can be developed on any C++ implementation compatible with the AT&T version 2.1 or above.

Readers with a knowledge of C should note that only Chapter 3, on the structure statements, contains no new features for a C programmer. There are small differences, as well as the more obvious larger ones, between the two languages. Such readers should scan Chapters 2, 4, 5, and 6 noting the smaller differences between the languages. Chapter 9 shows that C++ provides easier ways of allocating and deallocating memory dynamically than C. Some parts of the C language have been omitted, since they were not essential to this introductory overview of C++, even though they have been inherited by C++.

All implementations of C++ include ANSI C libraries. This gives a common core to all C++ programming. For this reason, the program examples and exercises use only these libraries, and deal only with numerical and text (character, string) handling.

Most screen-handling facilities, and specialist input facilities such as those for a mouse, are provided with further C++ and/or C libraries specific to the implementation. For example, Borland Turbo C++ provides ObjectWindows, which make use of the MicroSoft Windows environment, to enable a programmer to create a Windows interface for an application program.

Appendix 1 gives details of entering, compiling, linking, and making C++ programs using either Unix with GNU C++ or Borland Turbo C++ for PCs.

My thanks to my students, to my colleagues, J. A. Bermudez and Alan Lake, and to my publisher Malcolm J. Stewart at Macmillan, for their encouragement and corrections to this text. All the remaining mistakes are my own!

Jean Ettinger
University of Westminster
Email: ettingj@uk.ac.westminster