Macmillan Computer Science Series

- A. Abdellatif, J. Le Bihan, M. Limame, *Oracle - A User's Guide*
I. O. Angell, *High-resolution Computer Graphics Using C*
I. O. Angell and G. Griffith, *High-resolution Computer Graphics Using Pascal*
C. Bamford and P. Curran, *Data Structures, Files and Databases, second edition*
P. Beynon-Davies, *Database Systems*
P. Beynon-Davies, *Information Systems Development, second edition*
Linda E.M. Brackenbury, *Design of VLSI Systems - A Practical Introduction*
Alan Bradley, *Peripherals for Computer Systems*
P.C. Capon and P.J. Jinks, *Compiler Engineering Using Pascal*
B. S. Chalk, *Computer Organisation and Architecture*
Eric Davalo and Patrick Naïm, *Neural Networks*
Joyce Duncan, Lesley Rackley and Alexandria Walker, *SSADM in Practice*
D. England *et al.*, *A Sun User's Guide, second edition*
Jean Ettinger, *Programming in C++*
J.S. Florentin, *Microprogrammed Systems Design*
Michel Gauthier, *Ada - A Professional Course*
M.G. Hartley, M. Healey and P.G. Depledge, *Mini and Microcomputer Systems*
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*
Tom Manns and Michael Coleman, *Software Quality Assurance, second edition*
G.P. McKeown and V.J. Rayward-Smith, *Mathematical Foundations for Computing*
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*
J.P. Pardoe and M.J. King, *Object Oriented Programming Using C++
- An Introduction*
Pham Thu Quang and C. Chartier-Kastler, *MERISE in Practice*
Ian Pratt, *Artificial Intelligence*
F.D. Rolland, *Programming with VDM*
S. Skidmore, *Introducing Systems Analysis, second edition*
S. Skidmore, *Introducing Systems Analysis, second edition*
A.G. Sutcliffe, *Human-Computer Interface Design, second edition*
C.J. Theaker and G.R. Brookes, *Concepts of Operating Systems*
M. Thorin, *Real-time Transaction Processing*
D.J. Tudor and I.J. Tudor, *Systems Analysis and Design - A Comparison of Structured
Methods*
A.J. Tyrell, *Eiffel Object-Oriented Programming*

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*

Peter Grossman, *Discrete Mathematics for Computing*

H. Harper and A. Meadows, *GNVQ Advanced Information Technology*

Percy Mett, *Introduction to Computing*

P.D. Picton, *Neural Networks*

Tony Royce, *COBOL - An Introduction*

Tony Royce, *Structured COBOL - An Introduction*

Tony Royce, *C Programming*

Object Oriented Programming Using C++

An Introduction

John Pardoe and Melv King

© J.P. Pardoe and M.J. King 1997

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.

The authors have asserted their rights to be identified as the authors of this work in accordance with the Copyright, Designs and Patents Act 1988.

First published 1997 by
MACMILLAN PRESS LTD
Houndmills, Basingstoke, Hampshire RG21 6XS
and London
Companies and representatives
throughout the world

ISBN 978-0-333-69241-7   ISBN 978-1-349-14449-5 (eBook)
DOI 10.1007/978-1-349-14449-5

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

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

10 9 8 7 6 5 4 3 2 1
06 05 04 03 02 01 00 99 98 97

Contents

Preface	ix
1 Programming Concepts	1
1.1 Computer programs	1
1.2 The way a program uses the computer's memory	2
1.3 Input and output of data	4
1.4 Computer program development	6
1.5 The user requirements specification	6
1.6 The design stages	7
1.7 Detailed design using pseudo-code	7
1.8 The programming stage	8
1.9 Compiling, running and testing a program	8
1.10 Documenting and maintaining programs	9
1.11 Writing high-quality computer programs	10
1.12 Exercises	10
2 Object Oriented Programming Concepts	12
2.1 Concept of a model	12
2.2 Object classes and objects	13
2.3 Inheritance	15
2.4 Encapsulation	17
2.5 Polymorphism	17
2.6 Modelling a system using object classes	17
2.7 Exercises	18
3 A First Program	21
3.1 The model for our first program	21
3.2 The C++ code for our first program	22
3.3 Coding style	26
3.4 Pseudo-code for a sequence	27
3.5 Exercises	28

4	An Introduction to Inheritance	29
4.1	Using a header file	29
4.2	Inheriting code from our first program into a second program	31
4.3	Exercises	34
5	Arithmetic	36
5.1	Integers	36
5.2	Assignment statements and integer arithmetic	37
5.3	Real arithmetic	38
5.4	A model for a student's assessment in a subject	38
5.5	Using integer and real numbers, the <code>initialise</code> function	42
5.6	The display functions	44
5.7	Exercises	46
6	An Introduction to Selection	49
6.1	Further development of a model for a student's assessment in a subject	49
6.2	The <code>if</code> statement	49
6.3	The <code>if-else</code> statement	52
6.4	Relational operators	53
6.5	Nested selections	54
6.6	Compound statements	56
6.7	Exercises	57
7	Further Concepts for Selection	62
7.1	Further development of a model for a student's assessment in a subject	62
7.2	Logical operators	62
7.3	Assigning and comparing characters	66
7.4	The <code>switch</code> statement	67
7.5	Exercises	71

8 Repetition	74
8.1 Refining the input processes for the student's marks	74
8.2 The while statement	77
8.3 The for statement	80
8.4 The do-while statement	82
8.5 Exercises	85
9 Functions	88
9.1 The function call	88
9.2 The function definition	89
9.3 A revised version of the student marks object class	90
9.4 Call by value parameters	91
9.5 Function results	93
9.6 Call by reference parameters	96
9.7 When to use call by value and call by reference	101
9.8 Exercises	101
10 Constructors and Destructors	106
10.1 Constructors	106
10.2 Constructors with parameters	109
10.3 Destructors	111
10.4 Exercises	112
11 Introduction to Arrays	114
11.1 The need for arrays	114
11.2 Using an array	114
11.3 Exercises	121
12 Arrays of Objects	124
12.1 Using an array of objects	124
12.2 Aggregation	128
12.3 Exercises	129
13 Two-dimensional Arrays	132
13.1 Using a two-dimensional array	132
13.2 An alternative approach for entering the data	138
13.3 Exercises	140

14 Object Lifetime and Dynamic Objects	143
14.1 Object lifetime	143
14.2 The scope of variables	144
14.3 The scope of objects	145
14.4 Dynamic variables	147
14.5 Dynamic objects	149
14.6 Exercises	155
15 Streams and Files	161
15.1 Disk files	161
15.2 Streams	162
15.3 File streams	164
15.4 Object persistence	172
15.5 Exercises	173
16 Introduction to Polymorphism	177
16.1 What is polymorphism?	177
16.2 Operator overloading	177
16.3 Operator functions	179
16.4 Function overloading	181
16.5 Generic functions	184
16.6 Exercises	187
Appendix A : Object Classes Used in Examples	189
Appendix B : Solutions to Exercises	193
Appendix C : C++ Keywords and Operators	246
Appendix D : The ASCII Character Set	248
Index	249

Preface

Aim of the book

The increasing use of object oriented techniques for software development is now a fact of life.

- Increasingly large software projects must use object oriented techniques.
- It has been predicted that object oriented programming will provide the focus for all new software development.
- The concepts of software reuse are becoming increasingly important.

The aim of this book is to provide an introductory course in the use of object oriented computer programming using the C++ language. It has been especially designed to teach those with no previous experience of computer programming.

Many student programmers learn more traditional, procedural approaches to computer programming before tackling the quite different concepts found in object oriented programming; many students find this switch difficult. Hence the need for an introductory text that encourages students to understand basic programming concepts and techniques within an object oriented framework.

Other books in this area have tended to incorporate so much material, much of it difficult to assimilate, that the novice reader could well be discouraged. The temptation to include too much has therefore been resisted. Complex terminology has been eliminated or explained in simple terms.

What we have included are the basic concepts and terminology of object oriented programming that will enable the reader to 'get started' and build up the confidence to move on to more advanced texts.

Content of the book

The book develops the techniques of object oriented programming at the same time as gradually introducing the language features of C++. Procedural aspects that should be included in any introductory text, such as the use of structured programming techniques, have been incorporated.

Many straightforward examples are used to introduce and illustrate both new techniques and language features.

Chapter 1 covers basic concepts in order that the reader can appreciate what a computer program is. The development of a program is then put in context by describing briefly, in general terms, the various stages involved. Chapter 2 introduces the basic object oriented terminology and illustrates how object classes are used to model a very simple system.

Having introduced a C++ program in chapter 3, the important concept of inheritance and the use of a header file are covered in chapter 4.

Chapters 5 to 8 cover, within an object oriented context, facilities found in most procedural languages, such as basic data types, arithmetic and control constructs for selection and repetition. The need for, and the difference between, the constructs for both selection and repetition are emphasised.

Chapter 9 concentrates on the use of programmer-defined functions; concepts are introduced by referring to functions already used including those found in standard C++ libraries. The next chapter on the use of constructors and destructors is a natural development.

The need for arrays and their use is described in chapters 11 to 13. The concept of an array of objects enables the difference between inheritance and aggregation to be explained.

Scope rules and object lifetime are emphasised in chapter 14 before introducing the concept of pointers for dynamic variables and objects. Then chapter 15 covers the use of file streams in general and how they may be used to facilitate object persistence.

Finally, in chapter 16, we introduce some of the fundamental concepts of polymorphism and use appropriate C++ language constructs in straightforward illustrative examples.

Use of the book

Each chapter starts with the learning objectives for the chapter and concludes with a number of exercises. The chapters have been kept relatively short so that the reader can use the exercises to ensure that concepts are understood and techniques mastered in accordance with the objectives.

There are two types of exercise. The first type is a self-assessment question designed to reinforce major points by referring the reader back to specific examples in the chapter. The second type of exercise involves either amending one or more of the example programs or writing a new program. The reader is strongly advised to attempt all the exercises and for the second type to run the program and get it working. Sample solutions to all the exercises are given in appendix B.

An outline design, indicating how each function achieves a specific task, is provided in most of the examples and all the practical exercises; it is written in a form of structured English (pseudo-code). This emphasises to the reader the importance of understanding 'how to solve the problem' before 'coding in C++'. It also demonstrates how to write pseudo-code and provides appropriate help for the novice when attempting the practical exercises.

A model and corresponding object class for a student's performance (examination and practical mark) in a subject (module) is introduced in a simplified form in chapter 5. This model is developed and refined in subsequent chapters as additional features of object oriented programming are introduced. This case study approach promotes the learning process as well as providing an awareness of the reality of software systems development using object oriented programming. A diagram showing all the object classes that are eventually used and the relationship between them is given in appendix A. This diagram also indicates where each class is first introduced in the text enabling the reader to refer back, if necessary, for the full specification.

Acknowledgement

The authors are grateful to Neil Hepworth of South Bank University for his helpful and constructive comments made while reviewing this book.