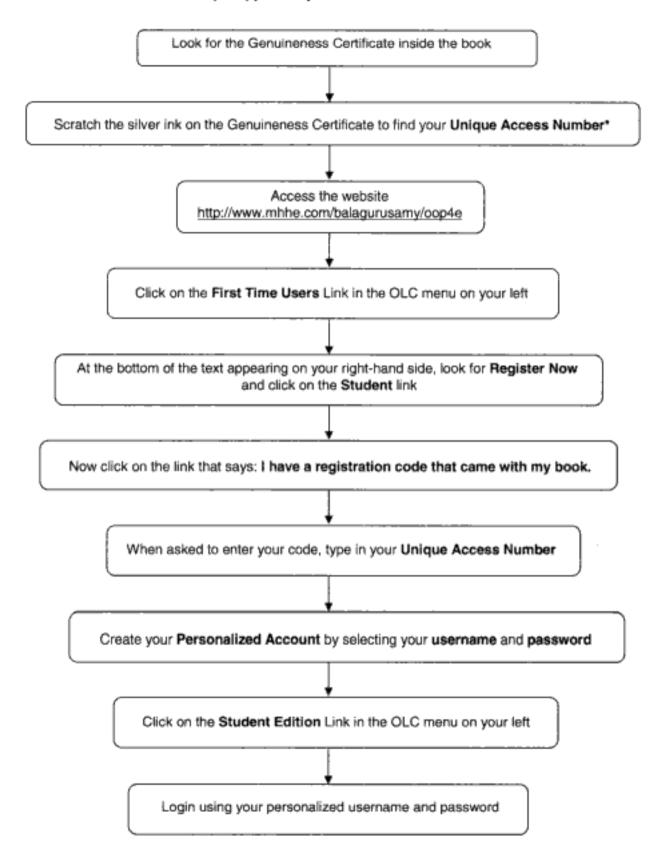
FOURTH EDITION



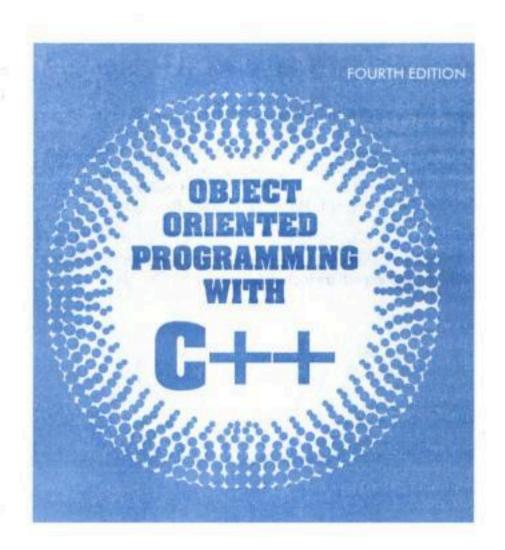
6-4-

E BALAGURUSAMY

Now a unique opportunity to access the Web Resources!



^{*} This number is meant for one time use and is self destructible



This One



About the Author

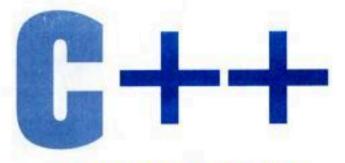
E Balagurusamy, former Vice Chancellor, Anna University, Chennai, is currently Member, Union Public Service Commission, New Delhi. He is a teacher, trainer, and consultant in the fields of Information Technology and Management. He holds an ME (Hons) in Electrical Engineering and a Ph. D. in Systems Engineering from the Indian Institute of Technology, Roorkee. His areas of interest include Object-Oriented Software Engineering, Electronic Business, Technology Management, Business Process Re-engineering, and Total Quality Management.

A prolific writer, he has authored a large number of research papers and several books. His best selling books, among others include:

- Programming in C#, 2/e
- Programming in Java, 3/e
- Programming in ANSI C, 4/e
- Programming in BASIC, 3/e
- Numerical Methods, and
- Reliability Engineering

A recipient of numerous honours and awards, he has been listed in the Directory of Who's Who of Intellectuals and in the Directory of Distinguished Leaders in Education.

OBJECT ORIENTED PROGRAMMING WITH



FOURTH EDITION

E Balagurusamy

Member Union Public Service Commission New Delhi



Tata McGraw-Hill Publishing Company Limited

NEW DELHI

McGraw-Hill Offices

New Delhi New York St Louis San Francisco Auckland Bogotá Caracas Kuala Lumpur Lisbon London Madrid Mexico City Milan Montreal San Juan Santiago Singapore Sydney Tokyo Toronto



Published by Tata McGraw-Hill Publishing Company Limited, 7 West Patel Nagar, New Delhi 110 008.

Copyright © 2008, 2006, 2001, 1994, by Tata McGraw-Hill Publishing Company Limited.

No part of this publication may be reproduced or distributed in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise or stored in a database or retrieval system without the prior written permission of the publishers. The program listings (if any) may be entered, stored and executed in a computer system, but they may not be reproduced for publication.

Fourth reprint 2008 DOLCRDRXRAZXB

This edition can be exported from India only by the publishers, Tata McGraw-Hill Publishing Company Limited.

ISBN (13 digits): 978-0-07-066907-9 ISBN (10 digits): 0-07-066907-4

Managing Director: Ajay Shukla

General Manager: Publishing-SEM & Tech Ed: Vibha Mahajan

Sponsoring Editor: Shalini Jha

Jr. Sponsoring Editor: Nilanjan Chakravarty

Senior Copy Editor: Dipika Dey

Senior Production Manager: P L Pandita

General Manager: Marketing-Higher Education & School: Michael J. Cruz

Product Manager: SEM & Tech Ed: Biju Ganesan

Controller—Production: Rajender P Ghansela Asst. General Manager—Production: B L Dogra

Information contained in this work has been obtained by Tata McGraw-Hill, from sources believed to be reliable. However, neither Tata McGraw-Hill nor its authors guarantee the accuracy or completeness of any information published herein, and neither Tata McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that Tata McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

Typeset at Script Makers, 19, A1-B, DDA Market, Paschim Vihar, New Delhi 110 063, and printed at Gopsons, A-2 & 3, Sector 64, Noida - 201 301

Cover: Gopsons

The McGraw-Hill Companies

Contents

| | Preface | |
|----|--|----|
| 1 | Principles of Object-Oriented Programming | 1. |
| | 1.1 Software Crisis 1 | |
| | 1.2 Software Evolution 3 | |
| | 1.3 A Look at Procedure-Oriented Programming 4 | |
| | 1.4 Object-Oriented Programming Paradigm 6 | |
| | 1.5 Basic Concepts of Object-Oriented Programming 7 | |
| | 1.6 Benefits of OOP 12 | |
| | 1.7 Object-Oriented Languages 13 | |
| | 1.8 Applications of OOP 14 | |
| | Summary 15 | |
| | Review Questions 17 | |
| 19 | Beginning with C++ | 2. |
| | 2.1 What is C++? 19 | |
| | 2.2 Applications of C++ 20 | |
| | 2.3 A Simple C++ Program 20 | |
| | 2.4 More C++ Statements 25 | |
| | 2.5 An Example with Class 28 | |
| | 2.6 Structure of C++ Program 29 | |
| | 2.6 Structure of C++ Program 29 2.7 Creating the Source File 30 | |
| | 2.8 Compiling and Linking 30 | |
| | Summary 31 | |
| | Review Questions 32 | |
| | Debugging Exercises 33 | |
| | Programming Exercises 34 | |
| 35 | Tokens, Expressions and Control Structures | 3. |
| | 3.1 Introduction 35 | |
| | 3.2 Tokens 36 | |
| | 3.3 Keywords 36 | |
| | 3.4 Identifiers and Constants 36 | |
| | | |
| | 3.6 User-Defined Data Types 40 | |
| | 3.7 Derived Data Types 42 | |
| | 3.1 Introduction 35 3.2 Tokens 36 3.3 Keywords 36 3.4 Identifiers and Constants 36 3.5 Basic Data Types 38 | 3. |

Contents -

| | 3.8 | Symbolic Constants 43 | |
|------------|------|--|----|
| | 3.9 | Type Compatibility 45 | |
| | 3.10 | Declaration of Variables 45 | |
| | 3.11 | Dynamic Initialization of Variables 46 | |
| | 3.12 | Reference Variables 47 | |
| | 3.13 | Operators in C++ 49 | |
| | 3.14 | Scope Resolution Operator 50 | |
| | 3.15 | Member Dereferencing Operators 52 | |
| | 3.16 | Memory Management Operators 52 | |
| | 3.17 | Manipulators 55 | |
| | 3.18 | Type Cast Operator 57 | |
| | 3.19 | Expressions and their Types 58 | |
| | 3.20 | Special Assignment Expressions 60 | |
| | 3.21 | Implicit Conversions 61 | |
| | 3.22 | Operator Overloading 63 | |
| | 3.23 | Operator Precedence 63 | |
| | 3.24 | Control Structures 64 | |
| | | Summary 69 | |
| | | Review Questions 71 | |
| | | Debugging Exercises 72 | |
| | | Programming Exercises 75 | |
| _ | | | |
| 4. | Fun | ections in C++ | 77 |
| ** | | | |
| | 4.1 | Introduction 77 | |
| | 4.2 | The Main Function 78 | |
| | 4.3 | Function Prototyping 79 | |
| | 4.4 | Call by Reference 81 | |
| | 4.5 | Return by Reference 82 | |
| | 4.6 | Inline Functions 82 | |
| | 4.7 | Default Arguments 84 | |
| | 4.8 | const Arguments 87 | |
| | 4.9 | Function Overloading 87 | |
| | 4.10 | Friend and Virtual Functions 89 | |
| | 4.11 | Math Library Functions 90 | |
| | | Summary 90 | |
| | | Review Questions 92 | |
| | | Debugging Exercises 93 | |
| | | Programming Exercises 95 | |
| | | a roge withing assertance or | |
| | - | t ott | |
| 5. | Clas | sses and Objects | 96 |
| Blanco III | | T . 1 .: 00 | |
| | 5.1 | Introduction 96 | |
| | 5.2 | C Structures Revisited 97 | |
| | 5.3 | Specifying a Class 99 | |

| | 5.6 | Making an Outside Function Inline 106 | |
|----|------|---|-----|
| | 5.7 | Nesting of Member Functions 107 | |
| | 5.8 | Private Member Functions 108 | |
| | 5.9 | Arrays within a Class 109 | |
| | 5.10 | Memory Allocation for Objects 114 | |
| | 5.11 | Static Data Members 115 | |
| | 5.12 | Static Member Functions 117 | |
| | 5.13 | Arrays of Objects 119 | |
| | 5.14 | Objects as Function Arguments 122 | |
| | 5.15 | Friendly Functions 124 | |
| | 5.16 | Returning Objects 130 | |
| | 5.17 | const Member Functions 132 | |
| | 5.18 | Pointers to Members 132 | |
| | 5.19 | Local Classes 134 | |
| | | Summary 135 | |
| | | Review Questions 136 | |
| | | Debugging Exercises 137 | |
| | | Programming Exercises 142 | |
| _ | 6.1 | Introduction 144 | |
| | 6.2 | Constructors 145 | |
| | 6.3 | Parameterized Constructors 146 | |
| | 6.4 | Multiple Constructors in a Class 150 | |
| | 6.5 | Constructors with Default Arguments 153 | |
| | 6.6 | Dynamic Initialization of Objects 153 | |
| | 6.7 | Copy Constructor 156 | |
| | 6.8 | Dynamic Constructors 158 | |
| | 6.9 | Constructing Two-dimensional Arrays 160 | |
| | 6.10 | const Objects 162 | |
| | 6.11 | Destructors 162 | |
| | | Summary 164 | |
| | | Review Questions 165 | |
| | | Debugging Exercises 166 | |
| | | Programming Exercises 169 | |
| | | | |
| 7. | Op | erator Overloading and Type Conversions | 171 |
| | 7.1 | Introduction 171 | |
| | 7.2 | Defining Operator Overloading 172 | |
| | | | |

Defining Member Functions 103

Overloading Unary Operators 173
Overloading Binary Operators 176

A C++ Program with Class 104

5.4

5.5

7.3 7.4

| VIII | • | Contents | |
|------|-------|--|-----|
| | 7.5 | Overloading Binary Operators Using Friends 179 | |
| | 7.6 | Manipulation of Strings Using Operators 183 | |
| | 7.7 | Rules for Overloading Operators 186 | |
| | 7.8 | Type Conversions 187 | |
| | **** | Summary 195 | |
| | | Review Questions 196 | |
| | | Debugging Exercises 197 | |
| | | Programming Exercises 200 | |
| 8. | Inh | neritance: Extending Classes | 201 |
| | 8.1 | Introduction 201 | |
| | 8.2 | Defining Derived Classes 202 | |
| | 8.3 | Single Inheritance 204 | |
| | 8.4 | Making a Private Member Inheritable 210 | |
| | 8.5 | Multilevel Inheritance 213 | |
| | 8.6 | Multiple Inheritance 218 | |
| | 8.7 | Hierarchical Inheritance 224 | |
| | 8.8 | Hybrid Inheritance 225 | |
| | 8.9 | Virtual Base Classes 228 | |
| | 8.10 | Abstract Classes 232 | |
| | 8.11 | Constructors in Derived Classes 232 | |
| | 8.12 | Member Classes: Nesting of Classes 240 | |
| | 0.12 | Summary 241 | |
| | | | |
| | | Review Questions 243 | |
| | | Debugging Exercises 243 | * |
| | | Programming Exercises 248 | |
| 9. | Poi | nters, Virtual Functions and Polymorphism | 251 |
| | 9.1 | Introduction 251 | |
| | 9.2 | Pointers 253 | |
| | 9.3 | Pointers to Objects 265 | |
| | 9.4 | this Pointer 270 | |
| | 9.5 | Pointers to Derived Classes 273 | |
| | 9.6 | Virtual Functions 275 | |
| | 9.7 | Pure Virtual Functions 281 | |
| | | Summary 282 | |
| | | Review Questions 283 | |
| | | Debugging Exercises 284 | |
| | | Programming Exercises 289 | |
| 10 | . M | anaging Console I/O Operations | 290 |
| _ | 10.1 | Introduction 200 | b |
| | 10.1 | Introduction 290 C++ Streams 291 | |
| | 141.4 | VITAMERIUS 401 | |

| | Contents | - 12 |
|-------|---|------|
| 10.3 | C++ Stream Classes 292 | |
| 10.4 | Unformatted I/O Operations 292 | |
| 10.5 | Formatted Console I/O Operations 301 | |
| 10.6 | Managing Output with Manipulators 312 | |
| | Summary 317 | |
| | Review Questions 319 | |
| | Debugging Exercises 320 | |
| | Programming Exercises 321 | |
| 1. Wo | orking with Files | 323 |
| 11.1 | Introduction 323 | |
| 11.2 | Classes for File Stream Operations 325 | |
| 11.3 | Opening and Closing a File 325 | |
| 11.4 | Detecting end-of-file 334 | |
| 11.5 | More about Open(): File Modes 334 | |
| 11.6 | File Pointers and Their Manipulations 335 | |
| 11.7 | Sequential Input and Output Operations 338 | |
| 11.8 | Updating a File: Random Acess 343 | |
| 11.9 | Error Handling During File Operations 348 | |
| 11.10 | Command-line Arguments 350 | |
| | Summary 353 | |
| | Review Questions 355 | 0 |
| | Debugging Exercises 356 | |
| | Programming Exercises 358 | |
| . Te | mplates | 359 |
| 12.1 | Introduction 359 | |
| 12.2 | Class Templates 360 | |
| 12.3 | Class Templates with Multiple Parameters 365 | |
| 12.4 | Function Templates 366 | |
| 12.5 | Function Templates with Multiple Parameters 371 | |
| 12.6 | Overloading of Template Functions 372 | |
| 12.7 | Member Function Templates 373 | |
| 12.8 | Non-Type Template Arguments 374 | |
| | Summary 375 | |
| | Review Questions 376 | |
| | Debugging Exercises 377 | |
| | Programming Exercises 379 | |
| . Ex | ception Handling | 380 |
| | | |

Introduction 380

Basics of Exception Handling 381

13.1 13.2 380

| X | • | Contents | |
|----|--------------|--|-----|
| | 13.3 | Exception Handling Mechanism 381 | |
| | 13.4 | Throwing Mechanism 386 | |
| | 13.5 | Catching Mechanism 386 | |
| | 13.6 | Rethrowing an Exception 391 | |
| | 13.7 | Specifying Exceptions 392 | |
| | | Summary 394 | |
| | | Review Questions 395 | |
| | | Debugging Exercises 396 | |
| | | Programming Exercises 400 | |
| 14 | Te | troduction to the Standard Template Library | 401 |
| | -2 | | 101 |
| | 14.1 | | |
| | 14.2 | Components of STL 402 | |
| | 14.3 | | |
| | 14.4 | Algorithms 406 Iterators 408 | |
| | 14.5 | | |
| | 14.6 14.7 | Application of Container Classes 409 | |
| | 14.7 | Function Objects 419 | |
| | | Summary 421 Bovious Overtions 422 | |
| | | Review Questions 423 Debugging Exercises 424 | |
| | | Programming Exercises 424 | |
| | | 1 Togramming Exercises 420 | |
| 15 | . м | anipulating Strings | 428 |
| | | | |
| | 15.1 | | |
| | 15.2 | Creating (string) Objects 430 | |
| | 15.3 | Manipulating String Objects 432 | |
| | 15.4 | Relational Operations 433 | |
| | 15.5 | String Characteristics 434 | |
| | 15.6 | Accessing Characters in Strings 436 | |
| | 15.7 | Comparing and Swapping 438 | |
| | | Summary 440 | |
| | | Review Questions 441 | |
| | | Debugging Exercises 442 | |
| | | Programming Exercises 445 | |
| 16 | . N | ew Features of ANSI C++ Standard | 446 |
| | 16.1 | Introduction 446 | |
| | 16.2 | New Data Types 447 | |
| | 16.3 | New Operators 449 | |
| | 16.4 | Class Implementation 451 | |

| | Debugging Exercises 464 Programming Exercises 467 | |
|----------------------|---|-------|
| 17. Ob | ject-Oriented Systems Development | 468 |
| 17.1 | Introduction 468 | |
| 17.2 | Procedure-Oriented Paradigms 469 | |
| 17.3 | Procedure-Oriented Development Tools 472 | |
| 17.4 | Object-Oriented Paradigm 473 | |
| 17.5 | Object-Oriented Notations and Graphs 475 | |
| 17.6 | Steps in Object-Oriented Analysis 479 | |
| 17.7 | Steps in Object-Oriented Design 483 | |
| 17.8 | Implementation 490 | |
| 17.9 | Prototyping Paradigm 490 | |
| 7.10 | Wrapping Up 491 | |
| | Summary 492 | |
| | Review Questions 494 | |
| Appendix A: Projects | | 496 |
| Appendix | B: Executing Turbo C++ | • 539 |
| Appendix | C: Executing C++ Under Windows | 552 |
| Appendix | D: Glossary of ANSI C++ Keywords | 564 |
| Appendix | E: C++ Operator Precedence | 570 |
| Appendix | F: Points to Remember | 572 |
| Appendix | G: Glossary of Important C++ and OOP Terms | 584 |
| Appendix | H: C++ Proficiency Test | 596 |
| Bibliograp | by | 632 |
| Index | | 633 |
| | | |

Namespace Scope 453 Operator Keywords 459

New Headers 461

Review Questions 463

Summary 461

16.5 16.6

16.7 16.8

1

Principles of Object-Oriented | Programming

Key Concepts

- Software evolution
- > Procedure-oriented programming
- Object-oriented programming
- Objects
- > Classes
- > Data abstraction
- Encapsulation
- Inheritance
- Polymorphism
- > Dynamic binding
- Message passing
- Object-oriented languages
- > Object-based languages

1.1 Software Crisis

Developments in software technology continue to be dynamic. New tools and techniques are announced in quick succession. This has forced the software engineers and industry to continuously look for new approaches to software design and development, and they are becoming more and more critical in view of the increasing complexity of software systems as well as the highly competitive nature of the industry. These rapid advances appear to have created a situation of crisis within the industry. The following issues need to be addressed to face this crisis:

- How to represent real-life entities of problems in system design?
- How to design systems with open interfaces?











