

THE WAITE GROUP

OBJECT-ORIENTED PROGRAMMING IN C++

SECOND EDITION

COVERS
Borland C++
Version 4.5
and Turbo C++

FULLY REVISED
EDITION OF THE
BEST-SELLING
OBJECT-ORIENTED
PROGRAMMING
IN TURBO C++

DISK
INCLUDED



WAITE
GROUP
PRESS

ROBERT LAFO

TABLE OF CONTENTS

| | |
|---|--------|
| Preface to the Second Edition | VII |
| Introduction | XXV |
| A Note to Teachers | XXVIII |
| Chapter 1 The Big Picture | 1 |
| Chapter 2 Using Borland C++ or Turbo C++ | 17 |
| Chapter 3 C++ Programming Basics..... | 41 |
| Chapter 4 Loops and Decisions | 81 |
| Chapter 5 Structures..... | 131 |
| Chapter 6 Functions..... | 161 |
| Chapter 7 Objects and Classes | 205 |
| Chapter 8 Arrays..... | 243 |
| Chapter 9 Operator Overloading..... | 285 |
| Chapter 10 Inheritance | 331 |
| Chapter 11 Borland C++ Graphics | 377 |
| Chapter 12 Pointers..... | 433 |
| Chapter 13 Virtual Functions and Other Subtleties | 491 |
| Chapter 14 Streams and Files..... | 549 |
| Chapter 15 Larger Programs | 609 |
| Chapter 16 Templates and Exceptions..... | 665 |
| Chapter 17 The Borland C++ Class Library | 703 |
| Chapter 18 Just for Fun | 753 |
| Appendix A ASCII Table | 783 |
| Appendix B Borland C++ Keywords..... | 787 |
| Appendix C Stand-Alone Graphics Programs | 788 |
| Appendix D Answers to Questions and Exercises..... | 791 |
| Appendix E Bibliography | 840 |
| Appendix F Debugging | 842 |
| | 846 |

| | |
|-------------------------------------|----|
| Custom Installation | 21 |
| Completing the Installation | 24 |
| Installing Turbo C++ | 24 |
| Full-Scale Installation | 24 |
| Custom Installation | 24 |
| Memory Models | 25 |
| Borland C++ Preliminaries | 26 |
| Starting Borland C++ | 26 |
| Projects | 27 |
| Turbo C++ Preliminaries | 29 |
| Your First Program | 30 |
| Naming Your Program | 30 |
| Using the Editor | 30 |
| Saving Your Program | 32 |
| Compiling and Linking | 32 |
| Running the Program | 34 |
| Exiting from the IDE | 35 |
| Files | 35 |
| Opening an Existing Project or File | 36 |
| Errors | 36 |
| Compiler Errors | 37 |
| Linker Errors | 37 |
| Run-time Errors | 37 |
| Conceptual Errors | 37 |
| DOS Programs in Borland C++ | 38 |
| Creating a DOS Executable | 38 |
| Running a DOS Executable | 39 |
| Summary | 39 |

Chapter 3 C++ Programming Basics

| | |
|----------------------------|----|
| Basic Program Construction | 43 |
| Functions | 43 |
| Program Statements | 44 |
| Whitespace | 45 |
| Output Using cout | 45 |
| String Constants | 46 |
| Preprocessor Directives | 46 |
| The #include Directive | |

| | |
|---|----|
| Integer Variables | 48 |
| Defining Integer Variables | 49 |
| Declarations and Definitions | 50 |
| Variable Names | 50 |
| Assignment Statements | 50 |
| Integer Constants | 51 |
| Output Variations | 51 |
| Character Variables | 51 |
| Character Constants | 52 |
| Initialization | 53 |
| Escape Sequences | 53 |
| Input with <code>cin</code> | 54 |
| Variables Defined at Point of Use | 54 |
| Cascading <code><<</code> | 55 |
| Expressions | 55 |
| Precedence | 55 |
| Type <code>float</code> | 56 |
| Floating-Point Constants | 57 |
| The <code>const</code> Qualifier | 57 |
| The <code>#define</code> Directive | 58 |
| Manipulators | 58 |
| The <code>endl</code> Manipulator | 58 |
| The <code>setw</code> Manipulator | 58 |
| Type <code>long</code> | 60 |
| Cascading the Insertion Operator | 60 |
| Multiple Definitions | 61 |
| The <code>IO Manip.H</code> Header File | 61 |
| Variable Type Summary | 61 |
| unsigned Data Types | 62 |
| Type Conversion | 64 |
| Automatic Conversions | 64 |
| Casts | 66 |
| Arithmetic Operators | 67 |
| The Remainder Operator | 67 |
| Arithmetic Assignment Operators | 67 |

| | |
|---------------------------------------|----|
| Increment Operators | 69 |
| Library Functions | 71 |
| Header Files | 72 |
| Library Files | 72 |
| Header Files and Library Files | 72 |
| Two Ways to Use <code>#include</code> | 74 |

| | |
|-----------|----|
| Summary | 74 |
| Questions | 75 |
| Exercises | 77 |

Chapter 4 Loops and Decisions

| | |
|---|----|
| Relational Operators | 83 |
| Loops | 84 |
| The <code>for</code> Loop | 84 |
| Debugging Features | 89 |
| <code>for</code> Loop Variations | 90 |
| The <code>while</code> Loop | 91 |
| Precedence: Arithmetic and Relational Operators | 94 |
| The <code>do</code> Loop | 95 |
| When to Use Which Loop | 97 |

| | |
|---|-----|
| Decisions | 97 |
| The <code>if</code> Statement | 98 |
| The <code>if...else</code> Statement | 101 |
| The <code>else...if</code> Construction | 108 |
| The <code>switch</code> Statement | 109 |
| The Conditional Operator | 113 |

| | |
|----------------------|-----|
| Logical Operators | 115 |
| Logical AND Operator | 116 |
| Logical OR Operator | 117 |
| Logical NOT Operator | 118 |

| | |
|-------------------------------------|-----|
| Precedence Summary | 118 |
| Other Control Statements | 119 |
| The <code>break</code> Statement | 119 |
| The <code>continue</code> Statement | 121 |
| The <code>goto</code> Statement | 123 |

| | |
|-----------|-----|
| Summary | 123 |
| Questions | 124 |
| Exercises | 126 |

Chapter 5 Structures

| | |
|------------|-----|
| Structures | 133 |
|------------|-----|

| | |
|-------------------------------|-----|
| A Simple Structure | 133 |
| Specifying the Structure | 134 |
| Defining a Structure Variable | 136 |
| Accessing Structure Members | 137 |
| Other Structure Features | 138 |
| A Measurement Example | 140 |
| Structures Within Structures | 142 |
| A Card-Game Example | 145 |
| Structures and Classes | 147 |
| Enumerated Data Types | 147 |
| Days of the Week | 148 |
| Inventing a Boolean Type | 150 |
| Organizing the Cards | 152 |
| Specifying Integer Values | 153 |
| Not Perfect | 154 |
| Other Examples | 154 |
| Summary | 154 |
| Questions | 155 |
| Exercises | 156 |

Chapter 6 Functions

| | |
|--|-----|
| Simple Functions | 164 |
| The Function Declaration | 164 |
| Calling the Function | 165 |
| The Function Definition | 165 |
| Comparison with Library Functions | 166 |
| Eliminating the Declaration | 167 |
| Passing Arguments to Functions | 168 |
| Passing Constants | 168 |
| Passing Variables | 169 |
| Passing by Value | 170 |
| Passing Structure Variables | 171 |
| Names in the Declaration | 173 |
| Returning Values from Functions | 174 |
| The <code>return</code> Statement | 175 |
| Returning Structure Variables | 177 |
| Reference Arguments | 179 |
| Passing Simple Data Types by Reference | 179 |
| A More Complex Pass by Reference | 181 |
| Passing Structures by Reference | 183 |

| | |
|--------------------------------|-----|
| Notes on Passing by Reference | 184 |
| Overloaded Functions | 184 |
| Different Numbers of Arguments | 185 |
| Different Kinds of Arguments | 186 |

| | |
|-------------------------------|-----|
| Inline Functions | 188 |
| Default Arguments | 191 |
| Variables and Storage Classes | 192 |
| Automatic Variables | 192 |
| External Variables | 194 |
| Static Variables | 196 |
| Storage | 197 |

| | |
|--|-----|
| Returning by Reference | 197 |
| Function Calls on the Left of the Equal Sign | 198 |
| Don't Worry Yet | 199 |

Summary 199

Questions 200

Exercises 202

Chapter 7 Objects and Classes

A Simple Class 207

Classes and Objects 208

Specifying the Class 208

Using the Class 211

C++ Objects As Physical Objects 213

C++ Objects As Data Types 214

Constructors 215

A Counter Example 216

Destructors 218

Objects As Function Arguments 218

Overloaded Constructors 220

The Default Copy Constructor 220

Member Functions Defined Outside the Class 222

Objects As Arguments 222

Returning Objects from Functions 224

How Many Arguments? 226

A Card-Game Example 228

Structures and Classes 230

Classes, Objects, and Memory 231

Static Class Data 232

Uses of Static Class Data 233