D.M. CAPPER

INTRODUCING C++

FOR SCIENTISTS, ENGINEERS AND MATHEMATICIANS

- *Learn C++, the highly popular object- oriented language, from scratch*
- *No prior knowledge of C required*
- *Skill level – undergraduate to professional*
- *Specifically aimed to meet the needs of scientists, engineers and mathematicians*

Springer-Verlag

2-005-541-1

2-005-541-1

D.M. Capper

C++ for Scientists, Engineers and Mathematicians

Springer-Verlag
London Berlin Heidelberg New York
Paris Tokyo Hong Kong
Barcelona Budapest

Contents

List of Figures xiii

List of Tables xv

Preface xvii

Structure of this Book . . . xviii

Acknowledgements . . . xx

1 Introduction 1

1.1 Getting Started . . . 1

1.2 Solving a Quadratic Equation . . . 3

1.3 An Object-oriented Example . . . 4

1.4 Why Object-oriented? . . . 6

1.5 Summary . . . 7

1.6 Exercises . . . 8

2 The Lexical Basis of C++ 9

2.1 Characters and Tokens . . . 9

2.2 Comments and White Space . . . 9

2.3 Identifiers . . . 11

2.4 Keywords . . . 12

2.5 Constants . . . 13

2.5.1 Integer Constants . . . 13

2.5.2 Floating Point Constants . . . 14

2.5.3 Character Constants . . . 14

2.5.4 String Constants . . . 15

2.6 Operators . . . 15

2.7 The Preprocessor . . . 16

2.8 Programming Style . . . 16

2.9 Summary . . . 17

2.10 Exercises . . . 18

3 Fundamental Types and Basic Operators 19
3.1 Integral Data Types 20
3.1.1 Type int 20
3.1.2 Integer Multiplication 23
3.1.3 Integer Division 23
3.1.4 Integer Modulus or Remainder Operator 24
3.1.5 Increment and Decrement Operators 25
3.1.6 Associativity and Precedence of Integer Operators 26
3.1.7 Long Integers 29
3.1.8 Short Integers 29
3.1.9 Unsigned Integers 30
3.1.10 Character Types 30
3.2 Floating Point Data Types 31
3.2.1 Type double 32
3.2.2 Type float 34
3.2.3 Type long double 34
3.3 Changing Types 35
3.3.1 Type Promotion and Conversion 35
3.3.2 Casts 37
3.4 Some Basic Operations 37
3.4.1 The sizeof Operator 38
3.4.2 Initialization 38
3.4.3 Assignment Operators 39
3.5 const 40
3.6 register and volatile 41
3.7 typedef 42
3.8 Summary 43
3.9 Exercises 43

4 Control Structure 45
4.1 Relational Operators 45
4.2 Logical Operators 47
4.3 Equal and Not Equal Operators 49
4.4 Blocks and Scope 50
4.5 Branch Statements 51
4.5.1 if Statement 51
4.5.2 if else statement 54
4.5.3 switch Statement 56
4.6 Iteration Statements 59
4.6.1 while Statement 59
4.6.2 for Statement 60
4.6.3 do Statement 62
4.7 break and continue Statements 64

4.8 goto Statement 65
4.9 Comma Operator 66
4.10 Null Statement 67
4.11 Conditional Expression Operator 68
4.12 Order of Evaluation of Operands 68
4.13 The Preprocessor 69
4.13.1 include Directive 70
4.13.2 define Directive 70
4.13.3 Conditional Compilation 72
4.14 Enumerations 74
4.15 Summary 76
4.16 Exercises 77

5 Functions **81**
5.1 Introducing Functions 81
5.1.1 Defining and Calling Functions 81
5.1.2 Return Type 84
5.1.3 Function Prototypes 86
5.1.4 Functions Cannot Declare Functions 86
5.1.5 Unused Arguments 88
5.1.6 Default Arguments 89
5.1.7 Ignoring the Return Value 91
5.2 Recursion 91
5.3 Inline Functions 93
5.4 More on Scope Rules 94
5.5 Storage Classes auto and static 95
5.6 Overloading Function Names 98
5.7 Function main() 100
5.8 Standard Library 101
5.9 Using Functions 104
5.9.1 A Benchmark 104
5.9.2 Root Finding by Bisection 106
5.10 Summary 110
5.11 Exercises 111

6 Pointers and Arrays **117**
6.1 Memory, Addressing and Pointers 117
6.1.1 'address-of' Operator 117
6.1.2 Dereferencing or Indirection Operator 119
6.1.3 Pointers 119
6.1.4 Pointers Do Not Declare Memory 123
6.1.5 Null Pointer 125
6.2 One-dimensional Arrays 125

6.2.1 Pointers and One-dimensional Arrays 127
6.2.2 Negative Array Indices 132
6.3 Type void* 132
6.4 Pointer Conversions 133
6.5 Multi-dimensional Arrays 133
6.5.1 Pointers and Multi-dimensional Arrays 135
6.6 Initializing Arrays 136
6.7 Size of Arrays 138
6.8 Arrays of Pointers and Multi-dimensional Arrays 139
6.9 Using Pointers and Arrays 141
6.9.1 Fitting Data to a Straight Line 141
6.9.2 Ragged Arrays 143
6.10 Summary 145
6.11 Exercises 148

7 Further Pointer Techniques **151**
7.1 Strings 151
7.2 Pointers as Function Arguments 154
7.3 Passing Arrays as Function Arguments 156
7.3.1 One-dimensional Arrays 156
7.3.2 Multi-dimensional Arrays 158
7.4 Arguments to main() 160
7.5 Pointers to Functions 163
7.6 Dynamic Memory Management 169
7.6.1 Allocating Memory 169
7.6.2 Deallocating Memory 175
7.7 Pass by Reference and Reference Variables 177
7.7.1 Reference Arguments 177
7.7.2 Reference Return Values 179
7.7.3 Reference Variables 181
7.8 Using Pointers, Arrays and Strings 182
7.8.1 Matrix Addition 182
7.8.2 An Alphabetic Sort 187
7.9 Summary 191
7.10 Exercises 193

8 Classes **195**
8.1 Declaring Classes 195
8.2 Class Access Specifiers 197
8.3 Accessing Members 198
8.4 Assigning Objects 200
8.5 Functions and Classes 201
8.6 Data Hiding 204

8.7 Returning an Object . . . 206
8.8 Reference Arguments . . . 207
8.9 Pointers to Members . . . 208
8.10 Pointer-to-member Operators . . . 210
8.11 Scope and Data Protection . . . 213
8.12 Static Members . . . 214
8.12.1 Static Data Members . . . 214
8.12.2 Static Member Functions . . . 216
8.13 Constructor Functions . . . 217
8.14 Constant Class Objects and Member Functions . . . 222
8.15 Friend Functions . . . 223
8.16 Program Structure and Style . . . 225
8.16.1 Modules . . . 225
8.16.2 Header Files . . . 228
8.17 Using Classes . . . 230
8.17.1 An Array Class . . . 230
8.17.2 A Weather Class . . . 235
8.18 Summary . . . 241
8.19 Exercises . . . 244

9 Operator Overloading 245
9.1 Introducing Overloaded Operators . . . 245
9.1.1 Overloading the Assignment Operator . . . 246
9.1.2 The this Pointer . . . 247
9.1.3 Overloading the Addition Operator . . . 249
9.1.4 Overloading the Unary Minus Operator . . . 251
9.2 User-defined Conversions . . . 252
9.2.1 Conversion by Constructors . . . 252
9.2.2 Conversion Functions . . . 253
9.2.3 Implicit Conversions . . . 257
9.3 Operator Function Calls . . . 259
9.3.1 Binary Operators . . . 260
9.3.2 Prefix Unary Operators . . . 262
9.3.3 Postfix Unary Operators . . . 263
9.4 Some Special Binary Operators . . . 264
9.4.1 Overloading the Subscript Operator . . . 264
9.4.2 Overloading the Function Call Operator . . . 266
9.5 Defining Overloaded Operators . . . 267
9.6 Using Overloaded Operators . . . 269
9.6.1 Complex Arithmetic . . . 269
9.6.2 Strings . . . 274
9.7 Summary . . . 280
9.8 Exercises . . . 282

10 Constructors and Destructors 283

10.1 More on Constructor Functions . . . 284

10.1.1 Dynamic Memory Management . . . 284

10.1.2 Assignment and Initialization . . . 285

10.1.3 Member Objects with Constructors . . . 287

10.2 Destructor Functions . . . 289

10.3 Creating and Destroying Objects . . . 290

10.4 Using Constructors and Destructors . . . 295

10.4.1 Singly Linked Lists . . . 295

10.4.2 Doubly Linked Lists . . . 302

10.5 Summary . . . 307

10.6 Exercises . . . 307

11 Single Inheritance 309

11.1 Derived Classes . . . 309

11.2 `virtual` Functions . . . 314

11.3 Abstract Classes and Pure `virtual` Functions . . . 318

11.4 Class Hierarchies . . . 322

11.5 Constructors and Destructors . . . 327

11.6 Member Access and Inheritance . . . 329

11.6.1 Access Specifiers . . . 329

11.7 Access Declarations . . . 331

11.7.1 Friendship and Derivation . . . 333

11.8 Using Single Inheritance . . . 335

11.8.1 A Bounds Checked Array Class . . . 335

11.8.2 A Menu Class . . . 340

11.9 Summary . . . 347

11.10Exercises . . . 351

12 Input and Output 353

12.1 Introduction . . . 353

12.2 Generic Input and Output Classes . . . 354

12.2.1 Output . . . 355

12.2.2 Input . . . 358

12.2.3 User-defined Types . . . 361

12.3 File Input and Output . . . 362

12.4 Formatting . . . 365

12.5 Stream Condition . . . 368

12.6 In Memory Input and Output . . . 370

12.7 Using the I/O Library . . . 372

12.8 Summary . . . 376

12.9 Exercises . . . 378

13 Bitwise Operations **381**
13.1 Bitwise Operators 381
13.1.1 Bitwise Complement 382
13.1.2 Bitwise AND 382
13.1.3 Bitwise Exclusive OR 382
13.1.4 Bitwise Inclusive OR 383
13.1.5 Shift Operators 383
13.1.6 Bitwise Assignment Operators 384
13.2 Bit-fields 385
13.3 Unions 386
13.4 Using Bitwise Operators 388
13.4.1 A Bit Array Class 388
13.4.2 The Sieve of Eratosthenes 394
13.4.3 Bit Representation of Integral Types 397
13.4.4 Bit Representation of Floating Point Types 399
13.5 Summary 403
13.6 Exercises 403

14 Multiple Inheritance **405**
14.1 Derived Classes 405
14.2 Virtual Base Classes 408
14.3 Constructors and Destructors 411
14.4 Member Access Ambiguities 412
14.5 Using Multiple Inheritance 414
14.6 Summary 424
14.7 Exercises 425

15 C++ Applications **427**
15.1 Finite Difference Techniques 427
15.1.1 Array Classes 429
15.1.2 Grid Class 432
15.1.3 Testing the Laplace Equation Solver 435
15.1.4 Exercises 440
15.2 A Simulation 440
15.2.1 Outline of Project 440
15.2.2 Graphics 441
15.2.3 Class Declarations 444
15.2.4 Utilities 450
15.2.5 Class Implementations 452
15.2.6 Exercises 468
15.3 Projects 469

Appendix–A Templates **471**

Appendix–B The ASCII Character Codes **475**

Appendix–C Operator Precedence and Associativity **477**

Appendix–D Differences between C and C++ **481**

Bibliography **485**

Index **489**