# Contents

*Preface*  

1. **Introduction**  
   1.1 A Simple Program  
      1.1.1 Executing the Program  
   1.2 Remarks  
      1.2.1 Execution Order  
   1.3 Repeating a Block of Commands  
      1.3.1 Drawing Any Regular Polygon  
      1.3.2 Repeat Within a Repeat  
   1.4 Some Useful Turtle Commands  
   1.5 Numerical Functions  
   1.6 Comments  
   1.7 Computation Without Graphics  
   1.8 Concluding Remarks  
      1.8.1 Graphics  
      1.8.2 A Note Regarding the Exercises  
   *Exercises*  

2. **A Bird’s Eye View**  
   2.1 Problem Solving Using Computers  
   2.2 Algorithms and Programs  
   2.3 Basic Principles of Digital Circuits  
   2.4 Number-Representation Formats  
      2.4.1 Unsigned Integer Representation  
      2.4.2 Signed Integers  
      2.4.3 Floating Point Representations  
   2.5 Organization of a Computer  
   2.6 Main Memory  
      2.6.1 Addresses  
      2.6.2 Ports and Operations  
   2.7 The Arithmetic Unit  
   2.8 Input-Output Devices  
      2.8.1 Keyboard  
      2.8.2 Display  
      2.8.3 Disks  
      2.8.4 Remarks  
   2.9 The Control Unit  
   2.10 The Timing Model  
   2.11 High-Level Programming Languages  
   2.12 Concluding Remarks  
   *Exercises*  

*Preface* xxii

1. **Introduction**  
2. **A Bird’s Eye View**  
3. **Exercises**
## 3. Variables and Data Types

- **3.1 Introduction**
  - **3.1.1 Identifiers**
  - **3.1.2 Literals and Variable Initialization**
  - **3.1.3 The const Keyword**
  - **3.1.4 Reading Data into a Variable**
  - **3.1.5 Printing**
  - **3.1.6 Exact Representational Parameters**

- **3.2 Arithmetic and Assignment**
  - **3.2.1 Integer Division and the Modulo Operator %**
  - **3.2.2 Subtleties**
  - **3.2.3 Overflow**
  - **3.2.4 Infinity and Not a Number**
  - **3.2.5 Explicit Type Conversion**
  - **3.2.6 Assignment Expression**

- **3.3 Examples**

- **3.4 Assignment with repeat**
  - **3.4.1 Programming Idioms**
  - **3.4.2 Combining Sequence Generation and Accumulation**

- **3.5 Some Operators Inspired by the Idioms**
  - **3.5.1 Increment and Decrement Operators**
  - **3.5.2 Compound Assignment Operators**

- **3.6 Blocks and Variable Definitions**
  - **3.6.1 Block**
  - **3.6.2 General Principle 1: Scope**
  - **3.6.3 General Principle 2: Shadowing**

- **3.7 Concluding Remarks**

**Exercises**

## 4. A Program-design Example

- **4.1 Specification**
  - **4.1.1 Examples**

- **4.2 Program Design**
  - **4.2.1 Testing**
  - **4.2.2 Correctness Proof**
  - **4.2.3 Invariants**

- **4.3 Debugging**

- **4.4 Comments in the Code**

- **4.5 Concluding Remarks**

**Exercises**

## 5. Simplecpp Graphics

- **5.1 Overview**
  - **5.1.1 y-axis Goes Downward!**

- **5.2 Multiple Turtles**

- **5.3 Other Shapes Besides Turtles**
  - **5.3.1 Circles**
5.3.2 Rectangles 63
5.3.3 Lines 63
5.3.4 Text 63

5.4 Commands Allowed on Shapes 64
5.4.1 Rotation in Radians 65
5.4.2 Tracking a Shape 65
5.4.3 Imprinting on the Canvas 65
5.4.4 Resetting a Shape 65

5.5 Clicking on the Canvas 66
5.6 Projectile Motion 66
5.7 Best-Fit Straight Line 67
5.8 Concluding Remarks 69

Exercises 69

6. Conditional Execution 71
6.1 The if Statement 71
6.2 Blocks 75
6.3 Other Forms of the if Statement 75
6.4 A Different Turtle Controller 78
   6.4.1 “Buttons” on the Canvas 78
6.5 The switch Statement 79
6.6 Conditional Expressions 81
6.7 Logical Data 82
   6.7.1 Reasoning About Logical Data 83
   6.7.2 Printing Bool Data 83
   6.7.3 Determining Whether a Number is Prime 84
6.8 Pitfalls 85
6.9 Concluding Remarks 86

Exercises 87

7. Loops 90
7.1 The while Statement 90
   7.1.1 Counting the Number of Digits 92
   7.1.2 Mark Averaging 93
7.2 The break Statement 95
7.3 The continue Statement 95
7.4 The do while Statement 96
7.5 The for Statement 97
   7.5.1 Variables Defined in initialization 98
   7.5.2 break and continue 99
   7.5.3 Style Issue 99
   7.5.4 Determining if a Number is Prime 99
7.6 Uncommon Ways of Using for 100
   7.6.1 Comma-separated Assignments 100
   7.6.2 Input In initialization and update 100
7.7 The Greatest Common Divisor 101
7.8 Correctness of Looping Programs 102
12.3 Default Values of Parameters  174
12.4 Function Overloading  174
12.5 Function Templates  175
  12.5.1 Function Templates and Header Files  176
12.6 Concluding Remarks  176
  Exercises  177

13. Practice of Programming: Some Tips and Tools  178
  13.1 Clarity of Specification  179
  13.2 Input-Output Examples and Testing  180
  13.3 Input/Output Redirection  181
  13.4 Design of Algorithms and Programs  182
    13.4.1 Mentally Execute the Program  184
    13.4.2 Test Cases for Code Coverage  184
  13.5 Assertions  184
    13.5.1 Disabling Assertions  185
  13.6 Debugging  186
    13.6.1 Debuggers and Ides  186
    13.6.2 End of File and Data-input Errors  187
    13.6.3 Aside: Input-Output Expressions  187
  13.7 Random Numbers  188
    13.7.1 The randuv Function In Simplecpp  189
  13.8 Concluding Remarks  189
  Exercises  189

14. Arrays  191
  14.1 Array: Collection of Variables  191
    14.1.1 Array-element Operations  192
    14.1.2 Acceptable Range for the Index  193
    14.1.3 Initializing Arrays  193
  14.2 Examples of Use  193
    14.2.1 Notation for Sub-arrays  194
    14.2.2 A Marks-display Program  194
    14.2.3 Who Got the Highest?  195
    14.2.4 General Roll Numbers  196
    14.2.5 Histogram  197
    14.2.6 A Taxi-dispatch Program  198
    14.2.7 A Geometric Problem  202
  14.3 The Inside Story  203
    14.3.1 Time Required to Access an Array Element  203
    14.3.2 Out-of-Range Array Indices  203
    14.3.3 The Array Name by Itself  205
    14.3.4 The Operator “[ ]”  205
  14.4 Function Calls Involving Arrays  206
    14.4.1 Examples  207
    14.4.2 Summary  208
  14.5 Selection Sort  208
## Contents

14.5.1 Estimate of Time Taken 210  
14.6 Representing Polynomials 210  
14.7 Array Length and const Values 211  
  14.7.1 Why const Declarations? 212  
  14.7.2 What We Use in this Book 212  
14.8 Concluding Remarks 213  
Exercises 213

15. More on Arrays 217  
  15.1 Character Strings 217  
    15.1.1 Output 218  
    15.1.2 Input 218  
    15.1.3 Character String Constant 219  
    15.1.4 Character-array Processing 220  
    15.1.5 Address Arithmetic 222  
  15.2 Two-Dimensional Arrays 222  
    15.2.1 Linear Simultaneous Equations 224  
    15.2.2 Passing Two-dimensional Arrays to Functions 225  
    15.2.3 Drawing Polygons in Simplecpp 226  
  15.3 Arrays of Pointers 227  
    15.3.1 Command-line Arguments to main 227  
  15.4 More Dimensions 228  
  15.5 Concluding Remarks 228  
Exercises 229

16. Arrays and Recursion 231  
  16.1 Binary Search 231  
    16.1.1 Estimate of Time Taken 233  
  16.2 Merge Sort 233  
    16.2.1 A Merging Algorithm 234  
    16.2.2 Mergesort Algorithm 234  
    16.2.3 Time Analysis 236  
  16.3 The 8 Queens Problem 236  
    16.3.1 Enforcing Constraints Early 239  
  16.4 Concluding Remarks 239  
Exercises 241

17. Object-oriented Programming: Structures 243  
  17.1 Basics of Structures 244  
    17.1.1 Visibility of Structure Types and Structure Variables 247  
    17.1.2 Arrays of Structures 247  
    17.1.3 Pointers to Structures and -> 247  
    17.1.4 Pointers as Structure Members 248  
    17.1.5 Linked Structures 248  
  17.2 Structures and Functions 249  
    17.2.1 const Reference Parameters 251  
    17.2.2 Passing Pointers to Structures 251  
  17.3 Taxi Dispatch Revisited 251
## Contents

### 17. Representing Vectors from Physics
- Representing Vectors from Physics 253

### 17. Member Functions
- Member Functions 255
  - Reference Parameters and `const` 256
  - Default Values to Parameters 257
  - The this Pointer 257
  - Capturing this in Lambda Expressions 257

### 17. Concluding Remarks
- Concluding Remarks 257
- Exercises 258

### 18. Object-oriented Programming: Classes
- Object-oriented Programming: Classes 260
  - Constructors 261
    - Calling the Constructor Explicitly 263
    - Default Values to Parameters 263
    - “Default” Constructor 264
    - Constructors of Nested Structures 264
    - Initialization Lists 265
    - Constant Members 265
  - The Copy Constructor 266
  - Destructors 267
  - Overloading Operators 267
  - Overloading Assignment 269
  - Access Control 270
    - Accessor and Mutator Functions 270
    - Prohibiting Certain Operations 271
  - Classes 271
  - Some Classes you have Already Used, Almost 272
    - Simplecpp Graphics 272
    - Standard Input and Output 272
  - Classes For File I/O 272
  - Header And Implementation Files 273
    - Separate Compilation 275
    - Remarks 275
  - Miscellaneous Features 275
    - Another Overloading Mechanism 275
    - Friends 276
    - Static Data Members 276
    - Static Member Functions 277
    - Template Classes 278
  - Concluding Remarks: The Philosophy of OOP 278
    - Exercises 279

### 19. A Project: Cosmological Simulation
- A Project: Cosmological Simulation 281
  - Mathematics of Cosmological Simulation 281
  - The Leapfrog Algorithm 283
  - Overview of the Program 285
    - Main Program 285
  - The Class `star` 287
Contents

19.5 Compiling and Execution  288
19.6 Concluding Remarks  288
   Exercises  289

20. Graphics Events and Frames  290
  20.1 Events  290
     20.1.1 Event Objects  290
     20.1.2 Waiting for Events  291
     20.1.3 Checking for Events  291
     20.1.4 Mouse Button-press Events  291
     20.1.5 Mouse Button Release Events  291
     20.1.6 Mouse-drag Events  292
     20.1.7 Key-press Events  292
  20.2 Frames  292
  20.3 A Drawing Program  292
  20.4 A Rudimentary snake Game  293
     20.4.1 Specification  293
     20.4.2 Classes  293
     20.4.3 The Main Program  295
  20.5 Concluding Remarks  295
     Exercises  296

21. Representing Variable-Length Entities  297
  21.1 The Heap Memory  298
     21.1.1 Accessibility and Lifetime  299
     21.1.2 A Worked Out Example  299
     21.1.3 Remarks  300
  21.2 Issues in Managing Heap Memory  301
  21.3 Representing Text  302
     21.3.1 Basic Principles  302
     21.3.2 Constructor  303
     21.3.3 The print Member Function  303
     21.3.4 Assignments  303
     21.3.5 Destructor  305
     21.3.6 Copy Constructor  305
     21.3.7 The [] Operator  306
     21.3.8 The Operator +  307
     21.3.9 Use  308
     21.3.10 A Class to Represent Students  309
  21.4 Concluding Remarks  310
     21.4.1 Class Invariants  310
     Exercises  310

22. The Standard Library  313
  22.1 The string Class  313
     22.1.1 Passing strings to Functions  315
  22.2 The Template Class vector  315
     22.2.1 Inserting and Deleting Elements  316
## Contents

### 22.2
- 22.2.2 The Type `size_t` 316
- 22.2.3 Index Bounds Checking 316
- 22.2.4 Functions on Vectors 317
- 22.2.5 Vectors of User-Defined Data Types 317
- 22.2.6 Multidimensional Vectors 318
- 22.2.7 A Matrix Class 318

### 22.3
- 22.3 Sorting a Vector 319
  - 22.3.1 Example 320
  - 22.3.2 Sorting an Array 321

### 22.4
- 22.4 The `map` Template Class 322
  - 22.4.1 Marks Display Again! 324
  - 22.4.2 Time to Access a Map 325

### 22.5
- 22.5 Containers and Iterators 325
  - 22.5.1 Finding and Deleting `map` Elements 328
  - 22.5.2 Inserting and Deleting `vector` Elements 328

### 22.6
- 22.6 Other Containers in the Standard Library 328

### 22.7
- 22.7 The `typedef` Statement 329
  - 22.7.1 More General Form 329

### 22.8
- 22.8 Remarks 330
- Exercises 330

### 23. Representing Networks of Entities 333
- 23.1 Graphs 333
- 23.2 Adjacency Lists 334
  - 23.2.1 Edges of Different Types 335
  - 23.2.2 Array/Vector Indices Rather Than Pointers 335
  - 23.2.3 Edges Represented Explicitly 336
- 23.3 Adjacency-Matrix Representation 336
- 23.4 Circuits 338
- 23.5 Surfing on the Internet 340
- 23.6 Concluding Remarks 342
- Exercises 342

### 24. Structural Recursion 343
- 24.1 Maintaining an Ordered Set 344
  - 24.1.1 A Search Tree 344
  - 24.1.2 Implementation 345
  - 24.1.3 On the Efficiency of Search Trees 347
  - 24.1.4 Packaging Search Trees/Sets 349
  - 24.1.5 Balancing a Search Tree 350
  - 24.1.6 Search Trees and `maps` 350
- 24.2 Layout of Mathematical Formulae 350
  - 24.2.1 Input Format 351
  - 24.2.2 Representing Mathematical Formulae 352
  - 24.2.3 Reading in a Formula 354
  - 24.2.4 The Drawing Algorithm 355
  - 24.2.5 Implementation 359
## 24.2.6 The Complete main program 361
24.2.7 Remarks 362
Exercises 362

### 25. Inheritance 365

#### 25.1 Turtles With an Odometer 366
   25.1.1 Implementation Using Composition 366
   25.1.2 Implementation Using Inheritance 367

#### 25.2 General Principles 368
   25.2.1 Access Rules and protected Members 369
   25.2.2 Constructors 371
   25.2.3 Destructors 371
   25.2.4 Other Operations on Subclass Objects 372
   25.2.5 The Type of a Subclass Object 372
   25.2.6 Assignments Mixing Superclass and Subclass Objects 372

#### 25.3 Polymorphism and Virtual Functions 373
   25.3.1 Virtual Destructor 375

#### 25.4 Program to Print Past Tense 375

#### 25.5 Abstract Classes 377

#### 25.6 Multiple Inheritance 378
   25.6.1 Diamond Inheritance 379

#### 25.7 Types of Inheritance 379

#### 25.8 Remarks 380
Exercises 380

### 26. Inheritance-based Design 382

#### 26.1 Formula-Drawing Revisited 383
   26.1.1 Basic Design 383
   26.1.2 Comparison of the Two Approaches 386
   26.1.3 Adding the Exponentiation Operator 386
   26.1.4 Reading in Formulae 387

#### 26.2 The Simplecpp Graphics System 388

#### 26.3 Composite Graphics Objects 390
   26.3.1 Ownership 391
   26.3.2 The composite Class Constructor 391

#### 26.4 A car Class 391

#### 26.5 Concluding Remarks 394
Exercises 395

### 27. Discrete-event Simulation 397

#### 27.1 Discrete-event Simulation Overview 398
   27.1.1 Discrete-time Systems 398
   27.1.2 Evolution of a Discrete time System 399
   27.1.3 Implementing a Discrete-time System 399
   27.1.4 Simple Examples of Use 402

#### 27.2 The Restaurant Simulation 404

#### 27.3 Resources 405
   27.3.1 A resource Class 405
27.3.2  Simple Example  406
27.3.3  The Coffee-Shop Simulation  407

27.4  Single-Source Shortest Path  408
27.4.1  Dijkstra’s Algorithm as a Simulation  411

27.5  Concluding Remarks  413
Exercises  413

28.  Simulation of an Airport  415
28.0.1  Chapter Outline  416

28.1  The Simulation Model  416
28.1.1  Overall Functioning  416
28.1.2  Safe Operation and Half-Runway Exclusion  417
28.1.3  Scheduling Strategy  417
28.1.4  Gate Allocation  418
28.1.5  Simulator Input and Output  418

28.2  Implementation Overview  418
28.2.1  Half-runway Exclusion  419
28.2.2  Gate Representation and Allocation  419

28.3  Main Program and Data Structures  419
28.4  The taxiway Class  420
28.5  The plane Class  422
28.5.1  Action land  425
28.5.2  Action requestAGate  426
28.5.3  Action turnToGate  426
28.5.4  Action atGate  427
28.5.5  Action backOnTaxiway  427
28.5.6  Action prepareToTakeOff  427
28.5.7  Action depart  427
28.5.8  Action for an Ordinary Segment  428

28.6  Deadlocks  428
28.7  Concluding Remarks  429
Exercises  429

29.  Systems of Non-Linear Equations  431
29.1  Newton–Raphson Method in Many Dimensions  432
29.2  The General Case  434
29.2.1  Termination  434
29.2.2  Initial Guess  434

29.3  How a Necklace Reposes  435
29.3.1  Formulation  435
29.3.2  Initial Guess  436
29.3.3  Experience  437

29.4  Remarks  437
Exercises  437

Appendix A  Simplecpp  438
A.1  Installation  438
A.2  Non-Graphics Features  438
| A.3 Graphics Classes and Functions | 439 |
| A.4 Compiling | 439 |

**Appendix B**  *Reserved Words in C++*  
**Appendix C**  *Operators and Operator Overloading*  
C.1 Bitwise Logical Operators 441  
C.1.1 OR 441  
C.1.2 AND 441  
C.1.3 Exclusive OR 442  
C.1.4 Complement 442  
C.1.5 Left Shift 442  
C.1.6 Right Shift 442  
C.2 Comma Operator 443  
C.3 Operator Overloading 443  

**Appendix D**  *Formatted Output*  
D.1 General-Purpose Manipulators 444  
D.2 Numeric Data 445  
D.2.1 Base 446  
D.2.2 Capitalization 446  
D.3 Boolean Data 446  

**Appendix E**  *The stringstream Class*  

**Appendix F**  *Exceptions*  
F.1 The General Form 450  
F.2 Enabling Input Exceptions 451  

**Appendix G**  *Managing Heap Memory*  
G.1 Reference Counting 453  
G.2 The Template Class `shared_ptr` 453  
G.2.1 Shared Pointers in the `string` Class 454  
G.2.2 General Strategy 455  
G.2.3 Shared Pointers in Expression Tees 456  
G.3 Weak Pointers 457  
G.3.1 Solution Idea 458  
G.4 Concluding Remarks 458  

**Appendix H**  *The C++ Preprocessor*  
H.1 Source-File Inclusion 459  
H.2 Macros 459  
H.2.1 Predefined Macros 460  
H.3 Conditional Inclusion 460  

**Index**  
463