Contents

1 Introduction **1**

2 A Survey of Programming Techniques **3**
2.1 Unstructured Programming 3
2.2 Procedural Programming 4
2.3 Modular Programming 5
2.4 An Example with Data Structures 5
2.4.1 Handling Single Lists 5
2.4.2 Handling Multiple Lists 7
2.5 Modular Programming Problems 7
2.5.1 Explicit Creation and Destruction 8
2.5.2 Decoupled Data and Operations 8
2.5.3 Missing Type Safety 9
2.5.4 Strategies and Representation 9
2.6 Object-Oriented Programming 10
2.7 Exercises 10

3 Abstract Data Types **13**
3.1 Handling Problems 13
3.2 Properties of Abstract Data Types 14
3.3 Generic Abstract Data Types 16
3.4 Notation 17
3.5 Abstract Data Types and Object-Orientation 18
3.6 Excercises 18

4 Object-Oriented Concepts **19**
4.1 Implementation of Abstract Data Types 19
4.2 Class 20
4.3 Object 21
4.4 Message 22
4.5 Summary 22
4.6 Exercises 23

5 More Object-Oriented Concepts **25**
5.1 Relationships 25
5.2 Inheritance 27
5.3 Multiple Inheritance 29
5.4 Abstract Classes 31
5.5 Exercises 32

6 Even More Object-Oriented Concepts **35**
6.1 Generic Types . . . 35
6.2 Static and Dynamic Binding . . . 37
6.3 Polymorphism . . . 38

7 Introduction to C++ **43**
7.1 The C Programming Language . . . 43
7.1.1 Data Types . . . 43
7.1.2 Statements . . . 44
7.1.3 Expressions and Operators . . . 46
7.1.4 Functions . . . 49
7.1.5 Pointers and Arrays . . . 50
7.1.6 A First Program . . . 51
7.2 What Next? . . . 52

8 From C To C++ **53**
8.1 Basic Extensions . . . 53
8.1.1 Data Types . . . 53
8.1.2 Functions . . . 55
8.2 First Object-oriented Extensions . . . 56
8.2.1 Classes and Objects . . . 56
8.2.2 Constructors . . . 58
8.2.3 Destructors . . . 60

9 More on C++ **63**
9.1 Inheritance . . . 63
9.1.1 Types of Inheritance . . . 63
9.1.2 Construction . . . 64
9.1.3 Destruction . . . 65
9.1.4 Multiple Inheritance . . . 65
9.2 Polymorphism . . . 66
9.3 Abstract Classes . . . 67
9.4 Operator Overloading . . . 68
9.5 Friends . . . 69
9.6 How to Write a Program . . . 70
9.6.1 Compilation Steps . . . 70
9.6.2 A Note about Style . . . 71
9.7 Excercises . . . 72

10 The List – A Case Study **73**
10.1 Generic Types (Templates) . . . 73
10.2 Shape and Traversal . . . 74
10.3 Properties of Singly Linked Lists . . . 75
10.4 Shape Implementation . . . 76
10.4.1 Node Templates . . . 76
10.4.2 List Templates . . . 78
10.5 Iterator Implementation . . . 80
10.6 Example Usage . . . 83
10.7 Discussion . . . 83
10.7.1 Separation of Shape and Access Strategies . . . 83
10.7.2 Iterators . . . 84
10.8 Exercises . . . 85

Bibliography **87**

A Solutions to the Exercises **89**
A.1 A Survey of Programming Techniques 89
A.2 Abstract Data Types 90
A.3 Object-Oriented Concepts 91
A.4 More Object-Oriented Concepts 92
A.5 More on C++ 94
A.6 The List – A Case Study 94