

Table of Content

Table of Content	i
Copyright.....	v
Dedication	vi
Preface.....	vi
Structure of This Book.....	vii
A Note on the Source Code.....	viii
Acknowledgments.....	viii
Where to Find More Information	ix
Typographical Conventions.....	ix
Chapter 1. Basic C++ Programming	1
1.1 How to Write a C++ Program	1
1.2 Defining and Initializing a Data Object	6
1.3 Writing Expressions	9
1.4 Writing Conditional and Loop Statements	13
1.5 How to Use Arrays and Vectors	19
1.6 Pointers Allow for Flexibility	23
1.7 Writing and Reading Files.....	26
Chapter 2. Procedural Programming.....	30
2.1 How to Write a Function	30
2.2 Invoking a Function.....	35
2.3 Providing Default Parameter Values	43
2.4 Using Local Static Objects	45
2.5 Declaring a Function Inline	47
2.6 Providing Overloaded Functions.....	48
2.7 Defining and Using Template Functions.....	49
2.8 Pointers to Functions Add Flexibility	52
2.9 Setting Up a Header File.....	54
Chapter 3. Generic Programming.....	57
3.1 The Arithmetic of Pointers.....	57
3.2 Making Sense of Iterators.....	62
3.3 Operations Common to All Containers	65
3.4 Using the Sequential Containers	66
3.5 Using the Generic Algorithms.....	69
3.6 How to Design a Generic Algorithm	71
3.7 Using a Map	77
3.8 Using a Set.....	78
3.9 How to Use Iterator Inserters.....	80
3.10 Using the iostream Iterators	81
Chapter 4. Object-Based Programming.....	85
4.1 How to Implement a Class	86
4.2 What Are Class Constructors and the Class Destructor?	89
4.3 What Are mutable and const?	94
4.4 What Is the this Pointer?.....	97
4.5 Static Class Members.....	99
4.6 Building an Iterator Class	102
4.7 Collaboration Sometimes Requires Friendship	106
4.8 Implementing a Copy Assignment Operator	108
4.9 Implementing a Function Object	109
4.10 Providing Class Instances of the iostream Operators	111
4.11 Pointers to Class Member Functions	112
Chapter 5. Object-Oriented Programming.....	117
5.1 Object-Oriented Programming Concepts.....	117

5.2 A Tour of Object-Oriented Programming	119
5.3 Polymorphism without Inheritance	123
5.4 Defining an Abstract Base Class	125
5.5 Defining a Derived Class.....	128
5.6 Using an Inheritance Hierarchy	133
5.7 How Abstract Should a Base Class Be?.....	135
5.8 Initialization, Destruction, and Copy	136
5.9 Defining a Derived Class Virtual Function	138
5.10 Run-Time Type Identification	141
Chapter 6. Programming with Templates	144
6.1 Parameterized Types.....	145
6.2 The Template Class Definition	147
6.3 Handling Template Type Parameters	148
6.4 Implementing the Template Class	150
6.5 A Function Template Output Operator	155
6.6 Constant Expressions and Default Parameters.....	156
6.7 Template Parameters as Strategy	160
6.8 Member Template Functions	161
Chapter 7. Exception Handling.....	164
7.1 Throwing an Exception	164
7.2 Catching an Exception.....	165
7.3 Trying for an Exception.....	167
7.4 Local Resource Management.....	170
7.5 The Standard Exceptions	172
Appendix A. Exercise Solutions	176
Exercise 1.4	176
Exercise 1.5	177
Exercise 1.6	179
Exercise 1.7	180
Exercise 1.8	181
Exercise 2.1	182
Exercise 2.2	183
Exercise 2.3	184
Exercise 2.4	185
Exercise 2.5	186
Exercise 2.6	187
Exercise 3.1	188
Exercise 3.2	190
Exercise 3.3	191
Exercise 3.4	194
Exercise 4.1	196
Exercise 4.2	197
Exercise 4.3	198
Exercise 4.4	199
Exercise 4.5	202
Exercise 5.1	205
Exercise 5.2	208
Exercise 5.3	209
Exercise 5.4	210
Exercise 6.1	210
Exercise 6.2	212
Exercise 7.1	216
7.2 Exercise 7.2.....	217
7.3 Exercise 7.3.....	218

Appendix B. Generic Algorithms Handbook.....	220
accumulate()	221
adjacent_difference()	221
adjacent_find().....	221
binary_search()	221
copy().....	222
copy_backward()	222
count().....	222
count_if()	222
equal().....	222
fill()	223
fill_n().....	223
find()	223
find_end()	223
find_first_of()	224
find_if().....	224
for_each().....	224
generate()	224
generate_n()	225
includes()	225
inner_product()	225
inplace_merge()	226
iter_swap()	226
lexicographical_compare()	226
max(), min().....	227
max_element() , min_element().....	227
merge().....	227
nth_element().....	228
partial_sort(), partial_sort_copy().....	228
partial_sum().....	229
partition(), stable_partition().....	229
random_shuffle()	229
remove(), remove_copy()	230
remove_if(), remove_copy_if()	230
replace(), replace_copy()	231
replace_if(), replace_copy_if()	231
reverse(), reverse_copy()	231
rotate(), rotate_copy()	231
search()	232
search_n()	232
set_difference()	233
set_intersection().....	233
set_symmetric_difference()	233
set_union()	233
sort(), stable_sort().....	234
transform()	234
unique(), unique_copy().....	235