

Contents

Preface	ix
0 Notes to the Reader	1
0.1 The structure of this book	2
0.2 A philosophy of teaching and learning	5
0.3 ISO standard C++	8
0.4 PPP support	11
0.5 Author biography	13
0.6 Bibliography	13
Part I: The Basics	
1 Hello, World!	17
1.1 Programs	18
1.2 The classic first program	18
1.3 Compilation	21
1.4 Linking	23
1.5 Programming environments	24

2 Objects, Types, and Values	29
2.1 Input	30
2.2 Variables	32
2.3 Input and type	33
2.4 Operations and operators	34
2.5 Assignment and initialization	36
2.6 Names	40
2.7 Types and objects	42
2.8 Type safety	43
2.9 Conversions	44
2.10 Type deduction: auto	46
3 Computation	51
3.1 Computation	52
3.2 Objectives and tools	53
3.3 Expressions	55
3.4 Statements	58
3.5 Functions	68
3.6 vector	71
3.7 Language features	77
4 Errors!	83
4.1 Introduction	84
4.2 Sources of errors	85
4.3 Compile-time errors	86
4.4 Link-time errors	88
4.5 Run-time errors	89
4.6 Exceptions	94
4.7 Avoiding and finding errors	99
5 Writing a Program	115
5.1 A problem	116
5.2 Thinking about the problem	116
5.3 Back to the calculator!	119
5.4 Back to the drawing board	126
5.5 Turning a grammar into code	130
5.6 Trying the first version	136
5.7 Trying the second version	140
5.8 Token streams	142
5.9 Program structure	146

6 Completing a Program	151
6.1 Introduction	152
6.2 Input and output	152
6.3 Error handling	154
6.4 Negative numbers	156
6.5 Remainder: %	157
6.6 Cleaning up the code	158
6.7 Recovering from errors	164
6.8 Variables	167
 7 Technicalities: Functions, etc.	 179
7.1 Technicalities	180
7.2 Declarations and definitions	181
7.3 Scope	186
7.4 Function call and return	190
7.5 Order of evaluation	206
7.6 Namespaces	209
7.7 Modules and headers	211
 8 Technicalities: Classes, etc.	 221
8.1 User-defined types	222
8.2 Classes and members	223
8.3 Interface and implementation	223
8.4 Evolving a class: Date	225
8.5 Enumerations	233
8.6 Operator overloading	236
8.7 Class interfaces	237
 Part II: Input and Output	
 9 Input and Output Streams	 251
9.1 Input and output	252
9.2 The I/O stream model	253
9.3 Files	254
9.4 I/O error handling	258
9.5 Reading a single value	261
9.6 User-defined output operators	266
9.7 User-defined input operators	266
9.8 A standard input loop	267

9.9	Reading a structured file	269
9.10	Formatting	276
9.11	String streams	283
10	A Display Model	289
10.1	Why graphics?	290
10.2	A display model	290
10.3	A first example	292
10.4	Using a GUI library	295
10.5	Coordinates	296
10.6	Shapes	297
10.7	Using Shape primitives	297
10.8	Getting the first example to run	309
11	Graphics Classes	315
11.1	Overview of graphics classes	316
11.2	Point and Line	317
11.3	Lines	320
11.4	Color	323
11.5	Line_style	325
11.6	Polylines	328
11.7	Closed shapes	333
11.8	Text	346
11.9	Mark	348
11.10	Image	350
12	Class Design	355
12.1	Design principles	356
12.2	Shape	360
12.3	Base and derived classes	367
12.4	Other Shape functions	375
12.5	Benefits of object-oriented programming	376
13	Graphing Functions and Data	381
13.1	Introduction	382
13.2	Graphing simple functions	382
13.3	Function	386
13.4	Axis	390

13.5	Approximation	392
13.6	Graphing data	397

14 Graphical User Interfaces 409

14.1	User-interface alternatives	410
14.2	The “Next” button	411
14.3	A simple window	412
14.4	Button and other Widgets	414
14.5	An example: drawing lines	419
14.6	Simple animation	426
14.7	Debugging GUI code	427

Part III: Data and Algorithms

15 Vector and Free Store 435

15.1	Introduction	436
15.2	vector basics	437
15.3	Memory, addresses, and pointers	439
15.4	Free store and pointers	442
15.5	Destructors	447
15.6	Access to elements	451
15.7	An example: lists	452
15.8	The this pointer	456

16 Arrays, Pointers, and References 463

16.1	Arrays	464
16.2	Pointers and references	468
16.3	C-style strings	471
16.4	Alternatives to pointer use	472
16.5	An example: palindromes	475

17 Essential Operations 483

17.1	Introduction	484
17.2	Access to elements	484
17.3	List initialization	486
17.4	Copying and moving	488
17.5	Essential operations	495

17.6	Other useful operations	500
17.7	Remaining Vector problems	502
17.8	Changing size	504
17.9	Our Vector so far	509
18	Templates and Exceptions	513
18.1	Templates	514
18.2	Generalizing Vector	522
18.3	Range checking and exceptions	525
18.4	Resources and exceptions	529
18.5	Resource-management pointers	537
19	Containers and Iterators	545
19.1	Storing and processing data	546
19.2	Sequences and iterators	552
19.3	Linked lists	555
19.4	Generalizing Vector yet again	560
19.5	An example: a simple text editor	566
19.6	vector , list , and string	572
20	Maps and Sets	577
20.1	Associative containers	578
20.2	map	578
20.3	unordered_map	585
20.4	Timing	586
20.5	set	589
20.6	Container overview	591
20.7	Ranges and iterators	597
21	Algorithms	603
21.1	Standard-library algorithms	604
21.2	Function objects	610
21.3	Numerical algorithms	614
21.4	Copying	619
21.5	Sorting and searching	620
	Index	625