
Contents

Preface, v

Chapter 1

Computing and the object-oriented design methodology	1
1.1 Basic computing terminology	2
Self-check questions	13
1.2 Software	18
1.3 Engineering software	24
1.4 Object-oriented design	32
Self-check questions	40
1.5 Points to remember	41
1.6 To delve further	43
1.7 Exercises	44

Chapter 2

C++: The fundamentals	49
2.1 Program organization	50
2.2 A first program	50
2.3 A second program	52
2.4 Comments	53
Self-check questions	56
2.5 Assigning a value	56
2.6 Fundamental C++ objects	57
2.7 Constants	61
Self-check questions	69
2.8 Names	69
2.9 Definitions	73
Self-check questions	76

2.10	Expressions	77
	Self-check questions	87
2.11	Output statements	88
2.12	Computing average velocity	91
	Self-check questions	93
2.13	Points to remember	95
2.14	Exercises	98

Chapter 3

	Modifying objects	103
3.1	Assignment	104
3.2	Const definitions	108
3.3	Input statements	109
	Self-check questions	111
3.4	Computing the number of molecules in a hydrocarbon	112
3.5	Compound assignment	115
3.6	Increment and decrement	117
	Self-check questions	119
3.7	Estimating yearly savings of change	120
3.8	The string class	122
	Self-check questions	128
3.9	EzWindows	130
3.10	Mowing lawns	134
	Self-check questions	141
3.11	Points to remember	141
3.12	Exercises	146

Chapter 4

	Control constructs	153
4.1	Boolean algebra	154
4.2	A Boolean type	156
	Self-check questions	161
4.3	Conditional execution using the if statement	164
	Self-check questions	173
4.4	Conditional execution using the switch statement	176
4.5	Computing a requested expression	179
4.6	Validating a date	181
	Self-check questions	188
4.7	Iteration using the while statement	189
4.8	Simple string and character processing	195
	Self-check questions	203
4.9	Iteration using the for construct	204

4.10	Simple data visualization	212
4.11	Solving the lazy hobo riddle	214
4.12	Iteration using the do construct	215
	Self-check questions	217
4.13	Points to remember	218
4.14	Exercises	222

Chapter 5

Function basics	233	
5.1	Function basics	234
5.2	The preprocessor	243
	Self-check questions	246
5.3	Using software libraries	246
5.4	The iostream library	248
5.5	The iomanip library	251
5.6	The fstream library	258
5.7	Random numbers	263
5.8	The assert library	268
	Self-check questions	269
5.9	Points to remember	272
5.10	To delve further	275
5.11	Exercises	275

Chapter 6

Programmer-defined functions	281	
6.1	Basics	282
6.2	A tasty problem	285
6.3	Some useful functions	290
6.4	Integrating a quadratic polynomial	293
6.5	The local scope	296
6.6	The global scope	299
	Self-check questions	301
6.7	Reference parameters	304
6.8	Passing objects by reference	312
6.9	Validating telephone access codes	314
	Self-check questions	317
6.10	Constant parameters	319
6.11	Default parameters	321
6.12	Casting of function parameters	323
6.13	Function overloading	324
	Self-check questions	326
6.14	Recursive functions	328
	Self-check questions	332

6.15	Displaying a price-interval stock chart	333
6.16	Points to Remember	342
6.17	To delve further	345
6.18	Exercises	345

Chapter 7

The class construct and object-oriented design	363	
7.1	Introducing a programmer-defined data type	364
7.2	The RectangleShape class	366
	Self-check questions	374
7.3	Using the RectangleShape class	375
7.4	Constructors	379
	Self-check questions	380
7.5	Building a kaleidoscope	382
7.6	Object-oriented analysis and design	387
7.7	Points to remember	397
7.8	To delve further	399
7.9	Exercises	400

Chapter 8

Implementing abstract data types	405	
8.1	Introducing abstract data types	406
8.2	Rational ADT basics	406
8.3	Rational interface description	413
	Self-check questions	421
8.4	Implementing the rational class	423
8.5	Copy construction, member assignment, and destruction	430
	Self-check questions	435
8.6	ADT for pseudorandom integers	437
8.7	Red-yellow-green game	444
8.8	Points to remember	461
8.9	Exercises	465

Chapter 9

Lists	471	
9.1	Named collections	472
9.2	One-dimensional arrays	472
	Self-check questions	483
9.3	Arrays as parameters	485

9.4	Sorting	488
	Self-check questions	493
9.5	Container classes	494
9.6	Class vector	497
9.7	QuickSort	511
9.8	Binary searching	516
9.9	String class revisited	518
	Self-check questions	520
9.10	Find that word—exploring a two-dimensional list	521
9.11	Maze runner	525
9.12	Multidimensional arrays	549
	Self-check questions	552
9.13	Points to remember	553
9.14	Exercises	558

Chapter 10

	The EzWindows API: a detailed examination	565
10.1	Application programmer interfaces	566
10.2	A simple window class	567
10.3	The bitmap class	577
10.4	Mouse events	579
10.5	Bitmaps and mouse events	583
10.6	Timer events	586
10.7	Alert messages	589
	Self-check questions	590
10.8	Simon says	591
10.9	Points to remember	607
10.10	Exercises	608

Chapter 11

	Pointers and dynamic memory	615
11.1	Lvalues and rvalues	616
11.2	Pointer basics	616
	Self-check questions	626
11.3	Constant pointers and pointers to constants	629
11.4	Arrays and pointers	630
11.5	Character string processing	633
11.6	Program command-line parameters	637
	Self-check questions	639
11.7	Pointers to functions	641
	Self-check questions	644
11.8	Dynamic objects	645

11.9	A simple ADT for representing lists of integer values	651
	Self-check questions	660
11.10	Points to remember	661
11.11	Exercises	665

Chapter 12

Testing and debugging	673	
12.1	Testing	674
	Self-check questions	682
	Self-check questions	693
12.2	Debugging	693
	Self-check questions	704
12.3	Points to remember	705
12.4	To Delve Further	705
12.5	Exercises	706

Chapter 13

Inheritance	707	
13.1	Object-oriented design using inheritance	708
13.2	Reuse via inheritance	709
13.3	A hierarchy of shapes	711
	Self-check questions	725
13.4	Protected members and inheritance	727
13.5	Controlling inheritance	730
	Self-check questions	732
13.6	Multiple inheritance	735
	Self-check questions	743
13.7	A prettier kaleidoscope	743
13.8	Points to remember	756
13.9	Exercises	759

Chapter 14

Templates and polymorphism	765	
14.1	Generic actions and types	766
14.2	Function templates	766
14.3	Class templates	769
14.4	A simple list class using a class template	770
	Self-check questions	778
14.5	Sequential lists	779
	Self-check questions	801

14.6	Polymorphism	801
14.7	Virtual function nuances	804
14.8	Abstract base classes	807
14.9	Virtual multiple inheritance	810
	Self-check questions	812
14.10	Points to remember	815
14.11	Exercises	817

Chapter 15

Software project–Bug Hunt!	823	
15.1	Bug hunt	824
15.2	Base class Bug	824
15.3	Class GameController	839
15.4	Bug hunt	842
	Self-check questions	843
15.5	Points to remember	844
15.6	Exercises	847

Appendix A

Tables	849	
A.1	ASCII character set	850
A.2	Operator precedence	851

Appendix B

Standard libraries	853	
B.1	Library naming and access	854
B.2	iostream library	854
B.3	Stdlib library	856
B.4	Math library	856
B.5	Time library	857
B.6	Cstring library	858
B.7	Algorithm library	860

Appendix C

Standard classes	865	
C.1	Container Classes	866
C.2	Class string	871

Appendix D

Advanced topics	875
D.1 Namespaces	876
D.2 Exception handling	881
D.3 Friends	888

Appendix E

EzWindows API reference manual	891
E.1 Enumerated types	892
E.2 Coordinate system	892
E.3 Class Position	894
E.4 Class SimpleWindow	894
E.5 Class WindowObject	898
E.6 Class RaySegment	898
E.7 Class Shape	900
E.8 Class EllipseShape	901
E.9 Class CircleShape	902
E.10 Class RectangleShape	902
E.11 Class TriangleShape	904
E.12 Class SquareShape	904
E.13 Class Label	905
E.14 Class BitMap	907
E.15 Class RandomInt	908
E.16 Miscellaneous functions	909

Appendix F

Projects and makefiles	911
F.1 Project and makefile fundamentals	912
F.2 Borland C++ IDE	912
F.3 Microsoft Visual C++ IDE	920
F.4 UNIX Makefiles	926

Index, 933