

C++ for Mathematicians

An Introduction for Students and Professionals

Edward Scheinerman

C

Chapman &. Hall/CRC

Taylor & Francis Group

Boca Raton London New York

Chapman & Hall/CRC is an imprint of the
Taylor & Francis Group, an informa business

Contents

List of Programs	xiii
List of Figures	xvii
Preface	xix
I Procedures	1
1 The Basics	3
1.1 What is C++?	3
1.2 Hello C++.	4
1.3 Exercises.	8
2 Numbers	11
2.1 The integer types.	11
2.2 The real number types.	14
2.3 The <code>bool</code> and <code>char</code> types.	14
2.4 Checking the size and capacity of the different types.	15
2.5 Standard operations.	18
2.6 Comparisons and Boolean operations.	22
2.7 Complex numbers.	23
2.8 Naming variables.	27
2.9 Exercises.	28
3 Greatest Common Divisor	31
3.1 The problem.	31
3.2 A first approach.	31
3.3 Euclid's method.	37
3.4 Looping with <code>for</code> , <code>while</code> , and <code>do</code> .	41
3.5 An exhaustive approach to the GCD problem.	43
3.6 Extended gcd, call by reference, and overloading.	45
3.7 Exercises.	49
4 Random Numbers	53
4.1 Pseudo random number generation.	53
4.2 Uniform random values.	54
4.3 More on pseudo random number generation.	57
4.4 A Monte Carlo program for the GCD problem.	60

4.5	Normal random values.	61
4.6	Exercises.	63
5	Arrays	67
5.1	Euler's totient.	67
5.2	Array fundamentals.	69
5.3	A procedure to factor integers.	71
5.4	A procedure to calculate Euler's totient.	76
5.5	The Sieve of Eratosthenes: <code>new</code> and <code>delete []</code>	78
5.6	A faster totient.	83
5.7	Computing p_n for large n	85
5.8	The answer.	87
5.9	Exercises.	88
II	Objects	91
6	Points in the Plane	93
6.1	Data and methods.	93
6.2	Declaring the <code>Point</code> class.	94
6.3	Data hiding.	97
6.4	Constructors.	99
6.5	Assignment and conversion.	100
6.6	Methods.	101
6.7	Procedures using arguments of type <code>Point</code>	103
6.8	Operators.	104
6.9	Exercises.	112
7	Pythagorean Triples	115
7.1	Generating Pythagorean triples.	115
7.2	Designing a primitive Pythagorean triple class.	116
7.3	Implementation of the <code>PTriple</code> class.	117
7.4	Finding and sorting the triples.	121
7.5	Exercises.	125
8	Containers	127
8.1	Sets.	127
8.2	Set iterators.	130
8.3	Multisets.	133
8.4	Adjustable arrays via the <code>vector</code> class.	134
8.5	Ordered pairs.	138
8.6	Maps.	139
8.7	Lists, stacks, and assorted queues.	144
8.7.1	Lists.	144
8.7.2	Stacks.	148
8.7.3	Queues.	148

8.7.4	Dequeues149
8.7.5	Priority queues150
8.8	Exercises.151
9	Modular Arithmetic	157
9.1	Designing the Mod type.157
9.2	The code.158
9.3	The default modulus: Static class variables and methods.163
9.4	Constructors and get/set methods.167
9.5	Comparison operators.167
9.6	Arithmetic operators.169
9.7	Writing Mod objects to output streams.172
9.8	A main to demonstrate the Mod class.172
9.9	Exercises.174
10	The Projective Plane	177
10.1	Introduction to the projective plane, MP^2177
10.2	Designing the classes PPoint and PLine.178
10.3	Inheritance.181
10.4	Protected class members.184
10.5	Class and file organization for PPoint and PLine.186
10.6	The parent class PObject187
10.7	The classes PPoint and PLine.195
10.8	Discovering and repairing a bug.200
10.9	Pappus revisited.207
10.10	Exercises.211
11	Permutations	215
11.1	Ulam's problem.215
11.2	Designing the Permutation class.217
11.2.1	Data <i>I</i>218
11.2.2	Constructors and destructors.218
11.2.3	Copy and assign220
11.2.4	Basic inspection and modification methods.223
11.2.5	Permutation operations.224
11.2.6	Comparison operators.225
11.2.7	Output.225
11.2.8	The code file Permutation, c.225
11.3	Finding monotone subsequences.229
11.4	Exercises.232

12	Polynomials	235
12.1	Procedure templates.	235
12.2	Class templates.	238
12.2.1	Using class templates.	238
12.2.2	Creating class templates.	239
12.3	The Polynomial class template.	242
12.3.1	Data	243
12.3.2	Constructors.	243
12.3.3	Get and set methods.	244
12.3.4	Function methods.	245
12.3.5	Equality.	246
12.3.6	Arithmetic.	246
12.3.7	Output to the screen.	247
12.3.8	GCD.	247
12.3.9	The code.	247
12.4	The GCD problem revisited.	254
12.5	Working in binary.	258
12.5.1	Signed versus unsigned integers.	258
12.5.2	Bit operations.	259
12.5.3	The bitset class template.	260
12.5.4	Class templates with non-type arguments.	263
12.6	Exercises.	264
III	Topics	267
13	Using Other Packages	269
13.1	Arbitrary precision arithmetic: The GMP package.	269
13.2	Linear algebra.	273
13.2.1	Two-dimensional arrays in C++.	273
13.2.2	The TNT and JAMA packages.	274
13.2.3	The newmat package.	282
13.3	Other packages.	286
13.4	Exercises.	287
14	Strings, Input/Output, and Visualization	289
14.1	Character arrays.	289
14.2	The string class.	291
14.2.1	Initialization.	291
14.2.2	Fundamental operations.	292
14.2.3	Searching.	295
14.2.4	Converting between string and char* types.	297
14.3	Command line arguments.	297
14.4	Reading and writing data in files.	300
14.4.1	Opening files for input/output.	300
14.4.2	Reading and writing.	303

14.4.3	Detecting the end of an input file	304
14.4.4	Other methods for input	305
14.5	String streams.	307
14.6	Formatting.	308
14.6.1	Setting precision.	309
14.6.2	Showing all digits.	309
14.6.3	Setting the width.	310
14.6.4	Other manipulators.	311
14.7	A class to parse files.	311
14.8	Visualization.	315
14.8.1	Introducing and installing the <code>plotutils</code> package	316
14.8.2	Drawing with <code>plotutils</code> —a first example.	317
14.8.3	Pascal's triangle modulo two.	322
14.8.4	Tracing the motion of a point moving randomly in a triangle	324
14.8.5	Drawing Paley graphs.	326
14.9	Exercises.	330
15	Odds and Ends	333
15.1	The <code>switch</code> statement	333
15.2	Labels and the <code>goto</code> statement	336
15.3	Exception handling.	338
15.3.1	The basics of <code>try</code> , <code>throw</code> , and <code>catch</code>	338
15.3.2	Other features of the exception-handling system	342
15.4	Friends.	344
15.5	Other ways to create types.	347
15.5.1	Structures.	347
15.5.2	Enumerations.	348
15.5.3	Unions.	348
15.5.4	Using <code>typedef</code>	349
15.6	Pointers.	350
15.6.1	Pointer basics.	350
15.6.2	Dereferencing	351
15.6.3	Arrays and pointer arithmetic.	353
15.6.4	<code>new</code> and <code>delete</code> revisited ¹	355
15.6.5	Why use pointers?.	356
15.7	Exercises.	358
IV	Appendices	361
A	Your C++ Computing Environment	363
A. 1	Programming with a command window and a text editor....	363
A. 1.1	What you need and how to get it (for free).	364
A. 1.2	Editing program files.	365
A. 1.3	Compiling and running your program.	366
A. 1.4	Compiler options.	368

A.1.5	Introduction to make.	370
A.2	Programming with an integrated development environment	372
A.2.1	Visual C++ for Windows.	373
A.2.2	Xcode for Macintosh OS X.	376
A.3	General advice on debugging.	378
B	Documentation with Doxygen	381
B.1	Doxygen comments	381
B.1.1	Documenting files.	382
B.1.2	Documenting procedures.	382
B.1.3	Documenting classes, data, and methods.	383
B.2	Using Doxygen	386
B.2.1	Configuring Doxygen.	386
B.2.2	Running Doxygen.	389
B.2.3	More features.	389
C	C++ Reference	391
C.1	Variables and types.	391
C.1.1	Fundamental types.	391
C.1.2	Standard classes/templates.	391
C.1.3	Declaring variables.	392
C.1.4	Static variables and scope.	392
C.1.5	Constants and the keyword <code>const</code>	393
C.1.6	Arrays.	393
C.2	Operations.	394
C.2.1	Assignment.	394
C.2.2	Arithmetic.	394
C.2.3	Comparison operators.	394
C.2.4	Logical operators.	394
C.2.5	Bit operators.	395
C.2.6	Potpourri.	395
C.3	Control statements.	396
C.3.1	<code>if-else</code>	396
C.3.2	Looping: <code>for</code> , <code>while</code> , and <code>do</code>	396
C.3.3	<code>switch</code>	397
C.3.4	<code>goto</code>	398
C.3.5	Exceptions.	398
C.4	Procedures.	398
C.4.1	File organization.	399
C.4.2	Call by value versus call by reference.	399
C.4.3	Array (and pointer) arguments.	400
C.4.4	Default values for arguments.	400
C.4.5	Templates.	400
C.4.6	<code>inline</code> procedures.	401
C.5	Classes.	401

C.5.1	Overview and file organization.	401
C.5.2	Constructors and destructors.	402
C.5.3	Operators.	403
C.5.4	Copy and assign.	404
C.5.5	static data and methods.	405
C.5.6	this.	406
C.5.7	Friends.	406
C.5.8	Class templates.	407
C.5.9	Inheritance.	407
C.6	Standard functions.	408
C.6.1	Mathematical functions.	408
C.6.2	Mathematical constants.	411
C.6.3	Character procedures.	411
C.6.4	Other useful functions.	413

D	Answers	415
----------	----------------	------------

	Index	487
--	--------------	------------