

Contents

1	Introduction to the Book	3
1.1	Introduction to QuantStart	3
1.2	What is this book?	3
1.3	Who is this book for?	3
1.4	What are the prerequisites?	4
1.5	Software Requirements	4
1.6	Book Structure	5
1.7	What the book does not cover	6
1.8	Where to get help	6
2	Introduction to C++	7
2.1	A Brief History Of C++	7
2.2	The Advantages Of C++	7
2.3	The Disadvantages Of C++	8
2.4	The Different Components Of C++	8
2.5	C-Style Procedural Programming	9
2.6	Object-Oriented Programming	9
2.7	Generic Programming	10
2.8	The Standard Template Library (STL)	10
2.9	The Boost Libraries	11
2.10	C++ In Quantitative Analysis	11
2.11	C++ In This Course	12
3	Your First Quantitative Finance C++ Program	13
3.1	Components Of An Object-Oriented C++ Program	13
3.2	What Are Classes?	14
3.3	Constructors And Destructors	14
3.4	Selectors And Modifiers	15

3.5	European Vanilla Option Specification	16
3.6	VanillaOption Header File	17
3.7	VanillaOption Source File	20
3.8	Passing By Reference Or Value	25
4	Option Pay-Off Hierarchies and Inheritance	27
4.1	Inheritance In Quantitative Finance	27
4.2	Option Pay-off Hierarchies	28
4.3	Implenting The PayOff Base Class	29
4.4	PayOff Header File	29
4.5	PayOffCall Header	30
4.6	PayOffDoubleDigital Header	31
4.7	PayOffCall Source File	32
4.8	PayOffDoubleDigital Source File	32
4.9	Virtual Destructors	33
5	Generic Programming and Template Classes	37
5.1	Why Generic Programming?	37
5.2	Template Classes	37
5.3	A Matrix Template Class	38
5.4	Template Syntax and Declaration	38
5.5	Default Types in Templates	39
5.6	SimpleMatrix Declaration	40
5.7	SimpleMatrix Implementation	43
5.8	Generic Programming vs OOP	47
6	Introduction to the Standard Template Library	49
6.1	Components of the STL	49
6.2	Containers	50
6.2.1	Sequence Containers	50
6.2.2	Associative Containers	51
6.2.3	Container Adaptors	52
6.3	Iterators	52
6.3.1	Iterator Categories	52
6.3.2	Iterator Adaptors	53

6.3.3	Const Iterators	54
6.3.4	Iterator Traits	55
6.4	Algorithms	55
6.4.1	Algorithm Categories	55
6.4.2	Nonmodifying Algorithms	56
6.4.3	Modifying Algorithms	57
6.4.4	Removal Algorithms	58
6.4.5	Mutating Algorithms	59
6.4.6	Sorting Algorithms	59
6.4.7	Sorted Range Algorithms	60
6.4.8	Numeric Algorithms	61
6.5	C++11 STL	62
6.5.1	Containers	62
6.5.2	Algorithms	62
7	Function Objects	65
7.1	Function Pointers	66
7.2	C++ Function Objects (Functors)	68
8	Matrix Classes for Quantitative Finance	71
8.1	Custom Matrix Class Library	72
8.1.1	C++ STL Storage Mechanisms	72
8.1.2	Matrix Mathematical Operations	73
8.1.3	Full Declaration	75
8.1.4	The Source File	77
8.1.5	Allocation and Deallocation	77
8.1.6	Mathematical Operators Implementation	80
8.1.7	Full Source Implementation	85
8.1.8	Using the Matrix Class	92
8.2	External Matrix Libraries	93
8.2.1	Installation	94
8.2.2	Basic Usage	94
8.2.3	Basic Linear Algebra	95
8.2.4	Expression Templates	95
8.2.5	Matrix and Scalar Arithmetic	96

8.2.6	Matrix Transposition	98
8.2.7	Matrix/Matrix and Matrix/Vector Multiplication	99
8.2.8	Vector Operations	100
8.2.9	Reduction	101
8.2.10	Useful Features	102
8.2.11	Next Steps	102
9	Numerical Linear Algebra	103
9.1	Overview	103
9.2	LU Decomposition	104
9.2.1	Algorithm	104
9.2.2	Eigen C++ Implementation	105
9.3	Thomas Tridiagonal Matrix Algorithm	107
9.3.1	C++ Implementation	108
9.4	Cholesky Decomposition	112
9.4.1	Algorithm	113
9.4.2	Eigen Implementation	113
9.5	QR Decomposition	115
9.5.1	Algorithm	115
9.5.2	Eigen Implementation	117
10	European Options with Monte Carlo	119
10.1	Black-Scholes Analytic Pricing Formula	119
10.2	C++ Implementation	120
10.3	Risk Neutral Pricing of a European Vanilla Option	126
10.4	C++ Implementation	128
11	Calculating the “Greeks”	135
11.1	Analytic Formulae	136
11.2	Finite Difference Method	142
11.3	Monte Carlo Method	145
12	Asian/Path-Dependent Options with Monte Carlo	151
12.1	Overview of Asian Options	151
12.2	Asian Options - An Object Oriented Approach	152
12.3	Pay-Off Classes	154

12.4 Path Generation Header	157
12.5 Asian Option Classes	160
12.6 The Main Program	164
13 Implied Volatility	169
13.1 Motivation	169
13.2 Root-Finding Algorithms	170
13.3 Interval Bisection Method	170
13.3.1 Interval Bisection with Function Templates	171
13.3.2 C++ Implementation	171
13.4 Implied Volatility via Newton-Raphson	177
13.4.1 Newton-Raphson Method	177
13.4.2 Pointer to a Member Function	178
13.4.3 Implementing Newton-Raphson	179
14 Random Number Generation and Statistical Distributions	185
14.1 Overview	185
14.2 Random Number Generator Class Hierarchy	186
14.2.1 Linear Congruential Generators	189
14.2.2 Implementing a Linear Congruential Generator	190
14.2.3 Implementation of the Main Program	194
14.3 Statistical Distributions	195
14.3.1 Statistical Distribution Inheritance Hierarchy	196
14.3.2 Standard Normal Distribution Implementation	197
14.3.3 The Main Listing	202
15 Jump-Diffusion Models	205
15.1 Modelling Jump-Diffusion Processes	205
15.2 Prices of European Options under Jump-Diffusions	206
15.3 C++ Implementation	207
16 Stochastic Volatility	211
16.1 Motivation	211
16.2 Process for Correlated Path Generation	212
16.3 Cholesky Decomposition	213
16.4 C++ Implementation	213

16.5	Mathematical Model	219
16.6	Euler Discretisation	220
16.6.1	Correlated Asset Paths	222
16.6.2	Monte Carlo Algorithm	222
16.7	C++ Implementation	223
16.7.1	PayOff Class	223
16.7.2	Option Class	223
16.7.3	Statistics and CorrelatedSND Classes	225
16.7.4	HestonEuler Class	225
16.7.5	Main Program	227
17	Single Factor Black-Scholes with Finite Difference Methods	233
17.1	Black-Scholes PDE for a European Call Option	234
17.2	Finite Difference Discretisation	235
17.3	Implementation	236
17.3.1	PayOff Class	237
17.3.2	VanillaOption Class	239
17.3.3	PDE Classes	241
17.3.4	FDM Class	244
17.3.5	Main Implementation	252
17.4	Execution and Output	254