

Index

- abstract factory pattern 131, 132–3
- access control
 - design considerations 90–1
 - implementation 107–8, 115
 - multiple systems 91, 108
 - operational risk and 24
 - pricing tool 190–1
 - single sign-on system 91, 108
- accounting entries 29, 100
- accounting rules 29, 100
- accrual accounting 213
- adapter pattern 123
- aggregate views 57–8
- American options, exercising 46
- analytics group
 - functions 21, 27, 34
 - requirements 27
- application programming interface (API)
 - 213
 - see also* WinAPI
- architectural aspects, visibility of 88
- arrays 71
- asset and liability management (ALM)
 - definition 213
 - functions 27–8
- asset-backed security (ABS) 214
- asset-class perspective 54–5, 210
- audit trails
 - definition 214
 - implementation 9, 108–10
 - technical requirements 38, 109
 - trade life-cycle 43, 55, 90, 94
- back office 16, 29, 214
- back-to-back trades 50–1
- banks
 - activities 30
 - back office 16, 29, 214
 - front office 16–20, 35, 42, 107
 - functional separation policy 90
 - group dynamics in 199–200
 - middle office 16, 20–8, 58, 107
 - overview 15–30
 - size 30
 - support functions 29–30
- barrier options 46
- batch jobs 214
- batch processing
 - design considerations 26, 102–5
 - implementation 113, 116
 - intermediate data storage 105–6
 - pricing tool implementation 191
 - regular processes 103
 - segmentation 104–5
 - time pressures 89, 103
- binomial tree pricing model 51, 53–4,
80–3, 214
 - example code 83, 125–9
 - pricing tool implementation 146–7,
148–9
- Black–Scholes pricing formula 51, 53–4,
214–15
 - example code 125–9
- Blitz++ 169, 179
- books 49, 210

- Boost 179
- bottom-line view 60–1, 210
- bugs 67, 69, 111
- Business Continuity Plans 24
- business control unit *see* treasury control unit
- Business Process Management (BPM) systems 39, 115, 190, 225
- C++
 - functions 72–3
 - overloading 77–8
 - standard template library 71
 - strong typing 71
- callable instruments 46
- casting 71
- class 75, 76, 78
- collateral management 36–7
- collateralized debt obligation (CDO) 193–4, 215
- The Commanding Heights* (Yergin & Stanislaw) 201
- commodity indices 96
- compilation 67–8
- compile-link-run-debug cycle 67–9
- complex data
 - errors 152
 - visualization 150–4
- compliance and reporting group 27
- component object model (COM) 133–4, 215
- computer processors 66
- computing
 - background information 67–75
 - dependencies 69–70
 - development cycle 67–9
 - paradigms 83–4, 119–20
 - relationship with finance industry 205
 - textbooks 199, 200–1
- The Concepts and Practice of Mathematical Finance* (Joshi) 196, 198
- Confessions of an Economic Hit Man* (Perkins) 201–2
- conflicts
 - policy decisions and 10–11
 - risk management vs trading room 10, 28, 208
- const declaration 73
- constructor method 76
- continuity 88
- corporate communications department 30
- costs
 - custom design 5
 - support 6
 - vendor API 6
- counterparty credit risk 23, 36, 208
 - exchange traded derivatives 44
 - over-the-counter trades 44
 - replication by trades 50
- credit defaults swap (CDS) 24, 193, 215
- credit derivatives 24, 215–16
- credit exposure 23–4, 106
- credit line allocation 36–7
- credit replication 24, 50, 101, 106–7, 115, 216
- credit risk
 - back-to-back trades 50–1
 - merger with market risk 24, 193
 - operational risk compared 26
 - sources 23
- credit risk management (CRM)
 - business unit functions 21, 23–4, 208
 - definition 216
 - systems 106, 113
 - using credit ratings 193
- curve (data type 141, 168
 - visualization 151, 153, 154
- Cygwin 70, 150, 179
- data replication 2
- data structures 71
- data types 70–1
- databases
 - access control 73, 90
 - audit requirements 94
 - choice 93–4, 113
 - complex architecture 113–17
 - flat file 218
 - geographical constraints 2, 61
 - hierarchical structure 94
 - importance 112–13
 - keys 94
 - market data 96–7
 - object 93, 221
 - relational 93, 223
 - trade representation 93–4
- date data type 140–1, 169
- date vector data type 141, 151, 168
- debuggers 69
- Delta 218

- Delta equivalent 216
- Delta hedging 216
- dependency checking 69–70
- deployability
 - incremental 2
 - model-centric perspective and 52–3
- derivatives 217
- Design Patterns: Elements of Reusable Object-Oriented Software* (Gamma *et al*) 200
- design patterns 119, 133, 134, 222
 - abstract factory pattern 131, 132–3
 - adapter pattern 123
 - advantages 119
 - choice 120, 134
 - disadvantages 120
 - façade pattern 121–3
 - factory method pattern 131–2, 133, 134
 - singleton pattern 129–30, 134
 - use 121
 - visitor pattern 123–9
 - vs paradigms 120–1
- desk quants *see* quants
- desk risk management 17, 20
- development cycle 67–9
 - complex projects 69–70
 - dependency rules 70
 - timestamps 70
- DISLIN 176, 179, 213
- DLL *see* dynamic link library
- documentation
 - automatic generation 88
 - importance 8
 - lack of 88
 - source code 88, 179
 - user 111
- documentation management 27
- double data type 140
- double dispatch 124
- Doxygen 88, 179, 204
- dynamic link library (DLL) 68, 217

- economics textbooks 201–2
- economists 17, 18
- encapsulation 76, 77, 84, 217
- end-of-day (EOD) processing *see* batch processing
- end-of-day (EOD) rates 26
- equity 217
- equity indices 96

- European options, exercising 46
- Exceptional C++* (Sutter) 200
- exchange-traded derivatives 43, 217
 - life-cycle events 44–5
 - vs over-the-counter 44
- executable files 68
- exotics trading 1, 4
- extensibility 9, 89–90

- façade pattern 121–3, 137
- factory method pattern 131–2, 134
- fair value 217
- Fermi, Enrico 84
- Feynman, Richard 84
- finance department
 - post-trade activities 47
 - profit and cost centre view 61, 211
 - role 30
- financial institutions *see* banks
- FinCAD 147
- fixed income 218
- fixing sources 99
- fixings 99
- flat file database 218
- forwards 181–90, 218
- frequently asked questions (FAQ) 111
- front office 16
 - access levels 107
 - business units 17–20
 - structure 17
 - trade inception 35
- functional programming 79–83, 84
 - advantages 83
 - computational laziness 80, 85
 - declarations 79–80, 85
 - purity 79, 85
 - statelessness 79, 83, 85
 - uses 85
- functions 71–2, 76–7
 - arguments passed by reference 72–3
 - arguments passed by value 72
 - overloading 77–8, 84
- futures 218

- Gamma 218
- Gaussian distribution 218
- geographic considerations 2, 61
- globalization 193
- globally unique identifiers (GUIDs) 134
- GNU C compiler 68

- Greeks
 - computation 41–2, 103
 - definition 42, 218
 - pricing tool function 139
- grid architecture 103–5
 - examples 113, 116
 - statelessness and 83, 85
- group dynamics 199–200
- Haskell 80, 82–3
- hedging
 - decisions 1, 42–3
 - definition 219
 - speed requirement 4
- high level languages 67
- high-frequency trading 4–5, 102
- human resources department 30, 47
- inception events 35–8, 206, 219
- incremental deployability 2
- information technology (IT) department 29–30
 - support role 111
 - system-centric perspective 61–2, 210
- inheritance 77, 84, 219
- integer data type 70, 140
 - casting 71
- integrated development environments 67
 - compile-link-run sequence 69
 - Makefiles 70
- integration, legacy systems 6, 7
- intellectual property
 - pricing models 3, 33
 - products and processes 7
 - vended systems 3, 5, 33
- interest rate derivatives 219
- interest rate swap (IRS) 43
- interfaces (COM objects) 133–4
- key person risk 6, 7
- knock out with rebate 46
- legal department 30
- life-cycle events 43–5, 115, 219
- limits monitoring 23, 42, 106
- linking 68
- loops 71
- maintainability 8, 88
- mark-to-market 26, 103, 220
- market data 206
 - ad hoc changes 95
 - asset-class specificity 96
 - audit trail requirement 108
 - database design 96–7
 - implementation 116, 129–30
 - importance 95
 - live data 26, 96
 - perspectives 38
 - quantization errors 181–90
 - rates and volatility 95
 - representation 94–7, 113
 - as separate system 2, 26
 - setup 95, 96, 97
 - snapshots 95, 96
 - sources 2, 26
- market data server 130
- market efficiency 195–6
- market risk
 - definition 23
 - merger with credit risk 24, 193
 - mitigating 25
 - modelling 25
 - operational risk compared 25
- market risk management 220
 - aggregate view 57–8
 - analytics 21, 27
 - functions 21, 23, 26, 40, 41–2
 - requirements 15, 23
- market value accounting 220
- marshalling 220
- Mathematica 80
- matrix data type 140, 169–70
 - visualization 151, 153, 154
- maturity 46
- member functions *see* methods
- members 76
- mentoring 8, 88
- menus 141, 152–3, 169
- methods 76, 78
- middle office 16, 20
 - access levels 107
 - business units 20–8
 - process flow 21, 58
 - structure 21
- model-centric perspective 51–3, 55, 210
- models
 - definition 220
 - development 34
 - effect of modelling on 195–7

- validation 27, 34, 221
- vs systems 205–6
- modularity 8
 - inheritance and 77
- money, philosophy of 202
- Monte Carlo simulation 221
- Moore's law 67
- More Exceptional C++* (Sutter) 200, 201
- mortgage-backed security (MBS) 221

- nondeliverable forwards (NDFs) 181–90
- normal distribution 221
- notional value 221
- novation 47

- object databases 93, 221
- object file 68
- object-oriented programming (OOP)
 - 75–9
 - advantages 78–9, 84
 - class 75, 76, 78
 - constructor method 76
 - disadvantages 66, 79, 84–5
 - encapsulation 76, 77, 84, 217
 - functional purity 79, 85
 - inheritance 77, 84, 219
 - interfaces 77
 - members 76
 - methods 75, 76, 78
 - objects 75
 - overloading 77–8, 84
 - polymorphism 77, 84, 222
 - statefulness 79, 84–5
- objects 75
- operation codes (opcodes) 67
- operational risk
 - definition 24
 - design considerations 90–1
 - mitigating 25
 - modelling 25
- operational risk management
 - functions 24
 - requirements 26
- operators 78
 - overloading 78
- options, exercising 46
- Options, Futures, and Other Derivatives* (Hull) 197
- outsourcing 5, 7
 - see also vended systems
- over-leveraging 193
- over-the-counter (OTC) trades 43, 217, 222
 - life cycle events 44
 - vs exchange traded derivatives 44
- paradigms
 - in computing 83–4, 119–20
 - definition 49
 - vs patterns 120–1
 - see also trade perspectives
- parallelization 113
 - scalability and 89, 103–5
 - statelessness and 83, 85
- passwords 91, 107, 108
- patterns see design patterns
- Paul Wilmott on Quantitative Finance* (Wilmott) 198–9
- performance 9–10
 - scalability and 89
- performance incentives 41, 28–9, 47
- persistence
 - definition 222
 - trade data 2, 43, 93
- physical security 91
- platform developers see quantitative developers
- plotting 156
 - pricing tool function 156–9, 160, 177–8
- policy implementation, via conflict 10–11
- polymorphism 77, 78, 84, 222
- portfolio-level view 57–8, 211
- portfolios
 - definition 49, 210
 - implementation 57–8
 - pricing 73–5
- post-trade activities 47
- potential future exposure (PFE) 24
 - back-to-back trades 50
 - computation 24, 106
 - definition 37, 222
- pre-trade activities 34–5
- pricing models
 - development 1, 4, 34–5, 51
 - implementation 19–20, 125–9
 - as intellectual property 3, 33
 - reusability 1–2
 - see also quant library

- pricing tool
 - access control 190–1
 - batch processing 191
 - CD contents 203–4
 - common quant classes 167–70
 - complex parameter visualization 150–4
 - context-sensitive help 139, 150, 155, 171
 - create/edit model interface 146–7, 175
 - create/edit product interface 145–6, 174–5
 - data types supported 140–1
 - design goal 138
 - error checking 145
 - external packages 179
 - features 3, 138–9
 - finite difference engine 159, 161, 162, 176
 - function templates 147–50
 - future enhancements 180–91
 - implementation details 165–79
 - life-cycle management 190
 - limitations 3, 138, 180
 - main control interface 143–5, 173–4
 - Makefile 70
 - market data 181
 - message boxes 178, 179
 - plotting interface 156–9, 160, 176
 - pluggable pricing models 176–7
 - pricing interface 154–6, 176
 - program flow 141–3
 - reusable components 177–9
 - scope 137–8
 - security 190–1
 - sensitivities calculation 159, 161, 162, 176
 - source code documentation 179
 - tool-tips 139, 155, 156, 171
 - trade database 180
 - type converters 178–9
 - user interactions 165–6
 - user interfaces 141–61, 170–6
- privileged information 61
- processes
 - importance of 10–11
 - as intellectual property 7
- product control
 - functions 20–2, 26, 40–1, 208–9
 - requirements 15
- product-centric perspective 53–4, 55, 210
- Profit and Loss (P/L)
 - explanation 20, 22
 - realized 43, 99
 - unrealised 40–1, 99, 209
 - value-at-risk and 58–60
- program design 73–5
- programming languages
 - choice 66
 - functional 79–83
 - high level 67
 - lexical structure 70–3
 - multiple 133
 - object-oriented 75–9
 - strongly-typed 70, 71
- quant library 4, 51–2
 - implementation 97–8, 112
- quantitative developers 3, 4, 223
- functions 17, 19–20
 - importance of big picture 62
 - product-centric perspective 53–4, 210
 - semantic contexts 52
- quantitative finance textbooks 195, 197–9
- quantization errors 181–90
- quants 3–4, 223–4
 - functions 17, 19, 51–2, 209
 - importance of big picture 62, 211
 - model-centric perspective 51–3, 210
 - pre-trade activities 34–5
- queues
 - implementation 55, 57
 - paradigm 55, 58, 211
- rate fixing 102, 223
- rates management
 - access levels 108
 - functions 21, 26
- reference data 223
 - management 97, 116
- regulatory requirements
 - audits 8
 - compliance and reporting 27, 29, 41
 - globalization and 193
 - security issues 7, 61, 91
- relational databases
 - serialization 93
 - trade data storage 93, 223
- reliability 9
- reports
 - bottom line view 61

- compliance reporting 27, 41
- implementation 116
- risk reporting 41
- sensitivities 23
- reserve 223
- reusability
 - component object model and 133, 134
 - design patterns and 119
 - pricing models 1–2
 - source code 71, 119, 133
 - user interfaces 98
- Rho 218
- risk analysis 1, 41–2
- risk appetite 11, 28
- risk controllers 10, 28–9
- risk management
 - access levels 108
 - conflict with trading room 10, 28, 208
 - functions 17, 28, 208
 - hedging decisions 43
 - requirements 20
 - see also* market risk management
- risk management systems 6, 7
- robustness
 - design patterns and 134
 - requirement for 9
- run-time errors 68–9
- sales personnel
 - functions 17, 18
 - inception events and 35
 - pre-trade activities 34
- scalability 89–90
 - batch jobs and 89, 103
 - importance 9
 - performance and 89
 - via modularity 8
 - see also* extensibility
- scenario analysis
 - example 55, 56, 57
 - pricing tool function 139
 - as regular process 103
- security requirements 9, 90–1
 - design 107–10
 - regulatory requirements 7, 61, 91
 - see also* access control
- semantic errors 69
- senior management, bottom-line view
 - 60–1, 210
- sensitivities
 - computation 41–2, 103
 - definition 218
 - pricing tool function 159, 161, 162, 176
 - regular process 103
 - reporting 23
- serialization 93, 220
- settlement 45–6
 - delay 101, 102
 - implementation 100–2, 115
 - straight-through processing 101–2
 - workflow 101, 113
- settlement risk
 - definition 23, 224
 - minimizing 102
- settlement systems
 - integration issues 6, 7, 100–1
 - pull implementation 100, 101
 - push implementation 100, 101
- shared object files 68, 217
- singleton pattern 129–30, 134
- source code
 - compilation 67
 - documentation 88
 - reusability 71, 119, 133
- speed, requirement for 4–5, 10
- spot FX trading 4–5
- spot prices 96
- standard template library (STL) 71
- static data 26, 224
 - management 26, 97, 116
- status flags
 - queue management 55, 211
 - trade transformations 98–9, 108
- Stochastic Calculus for Finance* (Shreve) 198
- straight-through processing (STP) 101–2, 106
- streaming 220
- stress tests 23, 41, 208
- strongly-typed languages 70, 71
- structured products
 - cancellation 50, 94
 - definition 18, 224
 - pre-trade activities 34
- structurers 17, 18, 35
- structuring tools 18
- support 16
 - costs 6
 - functions 29–30

- support (*Cont.*)
 - strategy implementation 110–11
 - tasks 111
- swaps 99
- system-centric view 61–2, 210
- systemic risks 24

- tail risk analysis 41
- target redemption trades 46
- termination of trades 45–7
- textbooks
 - computing 199, 200–1
 - economics 201–2
 - quantitative finance 195, 197–9
- Theta 42, 218
- time sensitivity measures 103
- timestamps 70
- TinyXml 179
- trade control unit *see* treasury control unit
- trade data server 130
- trade identification numbers 37–8, 94
- trade perspectives 33–4, 48, 49
 - aggregate views 57–8
 - asset-class view 54–5, 210
 - big picture vs narrow 62
 - bottom-line view 60–1, 210
 - finance department 61, 211
 - IT team 61–2
 - model-centric view 51–3, 55, 210
 - portfolio-level view 57–8, 211
 - product-centric view 53–4, 55, 210
 - queues and status flags 55, 58, 211
 - regulatory view 61
 - system-centric view 61–2, 210
 - trade-centric view 49–50, 210
- trade-centric perspective 49–50, 210
- trades
 - ageing 103
 - booking process 9, 37, 90, 115
 - cancellation 2, 37, 94, 107
 - inception 35–8, 206, 219
 - information demanded 15
 - life cycle 15, 33–47, 206–10
 - life-cycle events 43–5, 115, 219
 - linked 94
 - modification 37, 107
 - modifications 108
 - persistence 2, 43, 93
 - post-trade activities 47
 - pre-booking approval 35–6
 - pre-trade activities 34–5
 - pricing 73–5, 206
 - queues 55, 211
 - reconstruction 90
 - regular processing 40–3, 208
 - representation 92–4, 113
 - settlement 45–6
 - as software objects 33, 47, 92–3
 - state of 55, 98–9, 108
 - termination 45–7
 - transformations 98–9, 108
 - validation 38–9, 208
- trading desks
 - asset class organisation 19, 54, 210
 - conflict with risk management 10, 28, 208
 - functions 10, 17, 19, 42
 - hierarchy 54
 - inception events and 35
 - requirements 19, 55
- trading platforms
 - accuracy and 22, 102
 - advanced architecture 113–17
 - collaboration requirement 65, 194
 - complex example 113–17
 - components 91–111
 - cost-effectiveness 1
 - current trends 7–8
 - definition 1–3
 - design considerations 8–10, 87–91
 - example architecture 111–13
 - flow business and 43, 44–5
 - implementation options 5–7
 - importance of big picture 33, 194, 210
 - in-house development 6–7, 8, 33, 209–10
 - incremental deployability 2
 - integration issues 6, 7
 - market data feeds 2, 26
 - as master systems 7
 - as model archive 1–2
 - modern approaches 54
 - operational risk management and 25, 26
 - outsourcing 5, 7
 - profit potential 7
 - quasi real-time feature 89
 - regular processing functions 3, 103
 - requirements 1–3, 15
 - speed 4–5
 - trade persistence 2, 43, 93

- typical functions 15
- user perspectives 62–3
- vended solutions 5, 33, 209
- vendor API use 5–6, 8
- transactional integrity 9
- treasury control unit
 - access levels 107
 - functions 21, 22
 - queue paradigm 55
 - trade validation 22, 38–9
- unified modelling language (UML) 224
- unrealised P/L 40–1, 99, 209
- user documentation 111
- user interfaces
 - access control 90
 - pricing tool 141–61, 170–6
 - trade transformations 98–9
- user profiles 107–8
- validation
 - by trading platform 22
 - implementation 98–9
 - models 27, 34, 221
 - queues 55, 58, 211
 - trades 22, 38–9, 208
- value at risk (VaR)
 - advantages and disadvantages 58–60
 - computation 23, 41, 59–60, 104–5, 208
 - definition 224
 - efficacy of 41
 - as regular process 103
- variables
 - declaration 70
 - in functional languages 79
 - scope 70
 - types 70
- VAX computers 68
- vector data type 140, 169–70
 - visualization 151, 153, 154
- Vega, definition 42
- vended systems 209
 - data sharing 97
 - functions 3, 15
 - intellectual property 3, 5, 33
 - settlement 100–1
 - as sub systems 7
- visitor pattern
 - disadvantages 129
 - example 125–9
 - features 123–4
- Visual Studio 68, 69, 70, 179
- visualization
 - complex data 150–1
 - pricing tool function 139, 156–9
 - requirements 177
- WinAPI 147, 170–1, 213
- workflow engines 39, 115, 190, 225
- World Wide Web 120
- XML serialization 93
- yield curves 181, 190, 225

<http://www.pbookshop.com>