Distance Learning Course Advanced C# for Computational Finance and Derivatives' Pricing

DATASIM EDUCATION BV

Part I: Essential C#

Overview of the .NET framework
- Type Safety and Memory Management
- Classes and Interfaces
- Namespaces and their use in .NET
- Common Language Runtime (CLR)

Object-Oriented Programming in C#
- Classes, fields and methods
- Pass-by-value and pass-by-reference
- Constructors and object initializers
- Structs
- Properties

Advanced Classes
- Inheritance
- Polymorphism and casting
- Virtual function members
- Abstract classes and abstract methods
- Boxing and unboxing

Interfaces and Components
- Interfaces and contracts
- Implementing interfaces; the different scenarios
- Should we use an abstract class or an interface?
- Why interfaces are essential for applications

Delegates
- Motivation: function pointers and callbacks
- Delegate types and delegate instances
- Writing *plug-in methods* with delegates
- Multi-cast delegates
- Delegates versus interfaces

Events
- Broadcasts and notification patterns
- Event accessors and modifiers
- Events and Windows programming
- Anonymous methods

Generics in C#
- Generic types
- Generic methods and parameters
- Generic delegates
- Genric containers for finance

Advanced Generics
- Generic constraints
- Subclassing of generic types
- Self-referencing generic declarations
- Generic events

Application in .NET, Part 1
- One-factor Monte Carlo option pricing
- Scoping the system
- System and component interfaces
- System Decomposition

Application in .NET, Part 2
- Choosing between interfaces and delegates
- Implementing classes
- Design and system patterns (*Builder*, *Mediator*)
- Creating a single-threaded solution
- Multi-threaded solution

Part II: Core Libraries and their Applications in Finance

Fundamental Data Structures
- Strings and text handling
- Dates and Times; time zone
- Financial dates: day-count convention
- Formatting and parsing
- Regular expressions

Collections
- Defining and iterating in containers
- The Array Class
- Linked lists
- Hash tables and sorted dictionaries

Creating your own Containers
- Combining inheritance and generics
- Composition as an alternative to inheritance
- Data structures: vectors and numeric matrices
- Multi-dimensional data structures
- Applications to finance

Streams and I/O
- Stream architecture
- Stream class' members
- File, memory and pipe streams
- Stream adapters

Serialization
- Serialization engines
- Data contract serializer
- Binary serializer
- XML serializer
- Creating and applying serializers

Design by Contract
- Background (Eiffel Programming language)
- Supplier and client: rights and responsibilities
- What is reliable and correct software?
- Correctness and Hoare triples
- Preconditions, postconditions and assertions
- Imperative and declarative statements

Code Contracts in .NET
- Overview of Code Contract
- The binary rewriter
- The Contract class
- Implementing contract by design
- Contracts on interfaces and abstract classes
- Dealing with contract failure

Disposal and Garbage Collection (GC)
- IDisposable, Dispose and Close
- Finalisers
- Automatic garbage collection
- GC internals
- Memory leaks

Part III: Design and Integration Techniques

Threads and Parallel Programming
- Introduction to multi-threading concepts
- Thread class
- Synchronisation
- Asynchronous delegates
- Locking
- Thread safety

Assemblies
- No more DLL hell!
- The structure of an assembly
- Private and shared assemblies
- Global Assembly Cache (GAC)
- Assemblies and their importance for component design

Reflection and Metadata
- What is reflection and why is it useful?
- Reflection applied to types
- Reflection applied to assemblies
- Reflection and member invocation
- Application configuration
- Dynamic code generation

Part IV: Creating Computational Finance Applications

Introduction to Excel and C# Integration
- From VBA to C#
- The Excel Object Model
- Using C# to access Excel

Advanced Excel and C# Applications
- Creating stand-alone Excel applications
- COM Add-ins
- Worksheet functions and Automation Add-ins
- Formulae in C#
- Two-way Data Interoperability (NumericMatrix/Range)
- Regex library

Excel Configuration
- Data access with XML, CSV and relational databases
- Multi-threading and Excel
- Using regular expressions and formulae

C# and Design Patterns
- Why design patterns and which ones to use
- Implementing patterns in C#
- Ready-made patterns in C#
- Applications in finance

Major Patterns
- Builder and Factory Method
- Adapter, Bridge
- Visitor, Strategy, Template Method
- Producer-Consumer Pattern and data feeds

PDE/FDM Option Solver
- Class diagram
- Migrating C++ application to C#
- Presentation in Excel
- Early exercise features

Excel-based Fixed Income Application
- Creating a structured product in C#
- Class design
- Bootstrapping and yield curves
- Linear and cubic spline interpolation
- Worksheet functions
- Add-ins

Integration with Legacy Code
- Calling native DLLs
- Marshalling classes and structs
- Callbacks from unmanaged code
- Shared memory issues
- Version control

Part V: Interoperability and Interfacing
Assemblies
- What are assemblies?
- Modules
- Assemblies versus regular .DLLs
- Assemblies for reuse, versioning, deployment and security
- Assembly contents: Metadata, Resources, code and manifest
- Private and shared assemblies
- Creating and using assembly DLLs
- Assemblies and namespaces
- Shared Assemblies and Versioning

Interfacing C++ and C#
- An introduction to C++/CLI
- Interoperability scenarios
- Using C# functionality from C++
- Wrapping C++ classes in C#
- Examples from Boost statistics library and Winforms

Part IV
The Monte Carlo Method in C#
- Stochastic Differential Equations (SDE)
- Geometric Brownian Motion (GBM)
- CEV model
- Stochastic volatility

Finite Difference Method for SDE
- Euler and Milstein method for GBM
- Predictor-corrector method
- QE method

Examples
- Short-rate
- Heston
- Jump models

Monte Carlo Engine in C#
- Modular decomposition
- Design of engine (Produce-consumer)
- Random number generators
- Parallel programming

Finite Difference Method (FDM)
- One-factor models
- Plain and barrier options
- Early exercise features
- The Crank Nicolson method
- Comparing FDM with trinomial method

Alternating Direction Explicit (ADE) Method
- Background and motivation
- ADE for one-factor models
- ADE for nonlinear pricing models
- Advantages of ADE

Two-Factor Model
- ADI and Splitting Methods
- Craig-Sneyd method
- Mixed derivatives and Janenko method
- ADE for two-factor problems

Overview of Bond and Fixed Income Pricing
- Bond Pricing: Design, Implementation and Excel Interfacing
- Overview of bonds and kinds of bonds
- Bond price and bond yield
- Convexity
- (Macauley) duration
- Accrued interest and dirty price
- Day count conventions

Short-term Interest Rate Futures and Options
- Introduction (short term interest rate futures and option description)
- Organizing and manage futures data and code
- Conventions for Liffe Futures
- Pricing Option
- Working Example: portfolio of options

Interest Rate Models
- Vasicek, CIR, Hull-White
- Exact solutions
- Approximate solutions: lattice, PDE, MC
- Calibration