Prof Matthew Fried
Spring2019CS@gmail.com

University of Connecticut, School of Business
Financial Modeling with C# (FNCE 5353)

Term: Spring 2019
Instructor: Matthew Fried
Contact email: Spring2019CS@gmail.com, Matthew.Fried@UCONN.edu

Course Goals:
The goal of this course is to introduce the student to financial models within the framework of a C# deployment. It is meant to fully prepare the student for a work environment. This course will cover the gamut of C#, giving the student fluency in programming financial applications. It will introduce modelling and data structures as well. Financial applications begin with simple interest rate calculations and progress through option pricing models. Applications focus on numerical methods. All code will be written in C#.

Course Prerequisites:
- Undergraduate-level mathematics
- Financial Risk Modeling
- Knowledge in financial institutions, markets and instruments

Grading:
The course includes several projects.

Course Text:
C# 6.0 and the .NET 4.6 Framework; Troelsen and Japikse
http://www.apress.com/us/book/9781484213339

Helpful secondary references/resources (latest editions):
a. Visual C# How to Program, Deitel, 6th edition, Pearson, 0-13-460154-8
b. *C# 7.0 in a Nutshell, Albahari, O'Reilly;* http://shop.oreilly.com/product/0636920083634.do
c. Visual C#; Gaddis
d. Financial Calculus, An Introduction to Derivative Pricing; Baxter
e. The Concepts and Practice of Mathematical Finance; Joshi
f. Principles of Finance with Excel; Benninga
g. Murach's C# 2015; Murach, Boehm
h. The C# Player's Guide (Second Edition); RB Whitaker (free online)

i. Microsoft Visual C# .NET Step by Step; Sharp
j. Design Patterns in C#; Metsker
k. Professional C# 6.0 and .NET Core 1.0; Christian Nagel
l. For Threading: http://www.albahari.com/threading/#_Introduction
m. C# in Depth; John Skeet
n. Numerical Analysis by Timothy Sauer
o. Numerical Methods in Finance and Economics; Brandimorte
p. Credit Risk Modeling using Excel and VBA; Loffler and Posch
q. Handbook of Monte Carlo Methods, Kroese
r. Monte Carlo Methods in Financial Engineering; Glasserman
s. Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling; Stewart

Course Structure:

The course is broken into three slightly overlapping units.

1. Introduction to C# - this includes a thorough walk-through of the major topics and concepts related to programming in C#. This covers the introduction to the IDE, decision statements and iterative statements, methods and arrays.
2. Advanced Topics and Data Structures - this includes practical programming procedures such as creating classes. It continues with powerful data structures such as linked lists and develops maps, trees, hashing and more.
3. Financial Applications - programming examples will include many financial applications, and, beyond this, the course will implement the above knowledge by modelling risk in various different computational/algorithmic/probabilistic design patterns as outlined below.

Week	Goals
1	● Introduction to C# (IDE, data-types, etc.), ● Variables and Operators ● Decision statements (if and switch) ● Loops (for and while) ● Simple financial applications such as compound interest, fowards, and bond pricing
2	● Methods ● Arrays ● Statistics ○ Expected Value ○ Standard Deviation ○ Skew and Kurtosis
3	● Classes (such as the BankAccount class, the Client class, the Derivative class, and the Portfolio class)

	● Objects ● Getters/Setters
4	● Inheritance/Interface/Polymorphism ● Generics/Collections ● Numerical Analysis Applications I ○ Floating point notation and the limits of calculation ○ Review of Taylor Series ○ Bisection method ○ Fixed point iteration ○ Newton's method
5	● Numerical Analysis Applications II ○ Gaussian elimination ○ LU factorization ○ PA=LU used for bond pricing ○ Iterative Methods: Jacobi and Gauss-Seidel ○ Positive definite matrices and Cholesky factorization ○ Var-Cov and Correlation matrices using above factorizations
6	● Data Structures ○ Lists ○ Stacks and Queues ○ Heaps ○ Trees and Graphs ● Advanced Data Structures ○ Recursion with Sorting ○ Hashing ○ Big-O Analysis ● Interpolation ○ Lagrange interpolation ○ Newton divided difference ○ Chebyshev interpolation
7	● I/O and Managing Data ● Errors and Exceptions ● Functional Programming with LINQ ● Linear Regression ● Non-Linear Regression ● Fitting data to the model
8	● Midterm
9	● Numerical Analysis III ○ QR Orthogonalization ○ Numerical Differentiation ■ Finite Difference

	■ Formulas ■ Romberg Integration and Quadrature
10	● Numerical Analysis IV ○ Basic review of ODEs ○ IV problems and solvers (with higher orders) ○ PRNGs and Monte Carlo ○ Power Laws ○ Discrete and Continuous Brownian Motion Applications
11	● Monte Carlo with Options ○ LCGs ○ Lattice structures ○ Sampling Methods ○ Monte Carlo models ○ Review of Variance Reduction techniques with applications to hedging ● Portfolio Optimization
12	● Multithreading ● Power Iteration methods ● SVD ● Unconstrained Optimization with Derivatives ○ Gradient Descent variations ● Introduction to SQL
13	● Markov Chain models ● Queueing Theory applications ● More SQL
14	● Presentations and Design Patterns

- I hope to spend time introducing/highlighting some of the differences between C# and C++, so that students will be able to code either.
- If there is time, interest rate models, machine learning techniques, fast matrix multiplication and inversion methods can be discussed as well.
- I hope to have an in depth discussion on Monte Carlo methods from Monte Carlo Methods in Financial Engineering by Glasserman.
- I hope to spend time developing Markov Chains based off of the approach by Stewart in Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling. Another free resource on a very related topic of queueing theory is on EdX: Queuing Theory: from Markov Chains to Multi-Server Systems.

Projects:

C# - Spring 2019
Prof Matthew Fried
Spring2019CS@gmail.com

There will be several “small” projects in the course. Throughout the course we will be building programs that will assist you in creating your project. Please always keep the projects in mind. It is essential that you not only learn specific applications, but that you are also able to collate this knowledge. I hope to give class time to assist you throughout the semester. More details will follow as the semester goes. You may work in groups of up to 5 students. All projects are due the last day of class - there will be presentations then.

1. This project will be a *Portfolio Evaluator*. The user will be able to input their equity and debt holdings as well as their other assets (house, CDs, etc.) and the Evaluator will:
    a. value the equity based on any valuation model you choose (i.e. they can tell you whatever info you like and you will tell them what it is actually worth, for instance, you may feel that company X is really worth more or less than the price per share and you should evaluate it as such).
    b. value the debt based on any valuation model you choose
    c. return the breakdown of the portfolio per asset, as a percentage of the whole.
    d. and return the expected return in 1, 5, 10, and 20 years *and other relevant statistics*.

2. This project will focus on implementation of data structures. The company you work for asks you to design a program that will allow them to record all equity trades and be able to search through them and analyze them as well. (Make sure to include comments. Feel free to make up data) Your program should:
    a. be able to store multiple levels of data in an intelligent way
    b. sort and search through the data (you should write your own sorting and searching algorithms, don’t use C# premade classes for this). You should have a:
        i. Insertion sort,
        ii. Another sort that YOU make (possibly quicksort OR heapsort),
    c. and give some basic statistics about the data

3. This project focuses on data manipulation. You are asked to analyze a swath of data.
    a. Save data into a file. The data should be 4 stock prices for a period of three years.
    b. Find the Cholesky of this data and use it to create random stock paths that have similar properties to your data. Find 20 random paths (using a linear congruential generator to get the random number) for each stock and find the average return for each stock. And price an option accordingly (calls and puts) for a strike of your choice.
    c. Create a program that solves PA=LU for any user given matrix
    d. Create a program that does least squares approximation and also solves for an exponential approximation. The program should be able to read in stock data from a file and give both equations (linear and exponential) as outputs.
    e. Use at least two other numerical methods to analyze your data as well.

4. This project will focus on evaluating risk metrics. (Make sure to include comments.) You will:
    a. upload equity data (feel free to make up or download existing data; about 100 data points is more than enough)
    b. analyze VaR, Vol, or some relevant measure of risk
    c. and create an options model (of your choice)
    d. Use the techniques we did with Monte Carlo simulation to analyze your data

Grading:

Projects/Presentation (all 4)	60%
MidTerm	25%
Udemy Course	10%
Attendance and Participation	5%

Udemy Course

Here is a link for a C# course on the Udemy.com:
https://www.udemy.com/c-sharp-and-financial-modeling/?couponCode=UCONN10

The course is a video presentation of the topics we cover. It is important that you review the videos, and it is imperative that you answer the questions and practice (although you don't need to hand anything in). As stated above, answering all the questions and doing the work there-within will be 15% of your grade.

Instructor Information:

Matthew Fried has an M.A. in Applied Mathematics from Queens College (CUNY). He worked in finance for several years, trading options for boutique hedge funds. He specializes in derivative modelling. He has run various tail-risk strategies for said funds and has privately managed portfolios as well. For the last several years he has been teaching in Queens College (CUNY) [adjunct] and Farmingdale State College (SUNY) [Visiting Professor]. He has taught advanced courses in math, finance, and computers.