**MF 703**

**Programming for Mathematical Finance**

**Fall 2020**

|  |  |  |
| --- | --- | --- |
| **Lectures:** | D1 Thursday, 4:00-6:45pm HAR322  E1 Thursday, 7:00-9:45pm HAR 324 | |
|  |  | |
| **Instructor:** | Professor Chris Kelliher | |
|  | Office: | Zoom |
|  | Email: | [cmk44@bu.edu](mailto:cmk44@bu.edu) |
|  |  | |
| **Office hours:** | Monday, 10:30-11:30 am (Zoom) | |
|  |  | |
| **Teaching assistants:** | George Kepertis (<kepertis@bu.edu>)  Nikos Vingos (<nvingos@bu.edu>) | |
|  |  | |
| **Course websites:** | <https://questromtools.bu.edu/portal/site/FALL20MF703D1>  <https://questromtools.bu.edu/portal/site/FALL20MF703E1> | |
|  |  | |
| **Objectives & Prerequisites:** | This is a graduate level programming course on object-oriented programming techniques and building quantitative finance software applications. We will spend the majority of our time writing code to solve real-world quantitative problems that I have encountered in my career.  This course is very applied and hands-on. You will be expected to write a significant amount of code both on your homework assignments and on your final project. I assume that you have taken multiple programming courses and are familiar with C++ and Python.  We will also deal mostly with applications related to quantitative finance. Because of this, I also expect you to be familiar with stochastic calculus, asset pricing, probability, statistics and time series analysis. | |
|  |  | |
| **Topics:** | We will cover the following programming topics:   * Working in a multi-programmer environment * Database Programming & Design Principals * Object Oriented Programming Techniques * Best practices for writing applications in Python & C++ * Techniques for developing generic, scalable software applications   We will also spend a significant amount of time building financial applications, including implementation of:   * Options Pricing Techniques * Monte Carlo Simulations * Matrix Decompositions * Finite Difference Schemes * Optimization Techniques * Algorithmic Trading Models * Risk Models | |
|  |  | |
| **Recommended literature:** | *How to Think Like a Computer Scientist with Python 3*  by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers  <http://openbookproject.net/thinkcs/python/english3e/>  page1image3755088  *C++ Programming*  by WikiBooks page1image3755504  <https://en.wikibooks.org/wiki/C%2B%2B_Programming>  It is not required that you purchase any books for this course, as the slides from the lectures should be sufficient. | |
|  |  | |
| **Grading weights:** | Final Exam: 35%  Course project: 35%  Homework: 30% | |
|  |  | |
| **Exam:** | There will be a take home exam on Thursday November 5th. Materials, such as books and coding references will be allowed during the exam, however, collaboration is prohibited and considered cheating. | |
|  |  | |
| **Course Project:** | There will be a project to be submitted on the last day of class. You will be required to build teams of 3 to 5 students to work with on a project of practical relevance.  The project will require you to go through the full lifecycle of a quantitative software engineering project from initial data collection and storage to model implementation to delivering the model results.  The project will give you a chance to research a topic that interests you. It is expected that you will create a robust implementation of the method/technique you are researching, along with some results of its application to a real-world problem. Your implementation should be done in C++ or Python.  A significant portion of the grade for this project will depend on the quality of the code that you submit. While the results are obviously important, you are not expected to do ground-breaking research and my focus on grading the projects will be the research process that you follow, and the quality of your implementation.  You are required to submit a project proposal by October 15th. The proposal should consist of a summary of what you are planning to research, an implementation plan, as well as multiple references that you plan to leverage in your work.  I will give you a great deal of flexibility in choosing your topic; however, it should be directly relevant to the topics discussed in class. I may provide feedback on your topic after your proposal is submitted.  I will also periodically meet with each group to discuss your progress and make sure you are on the right track. It is expected that you will come prepared to these meetings and be ready to discuss your progress.  You are required to submit a written summary of at most 10 pages containing your method and results. You are also required to submit all code that you wrote.  Also, your team will be required to give a 10-minute presentation in class during the last week of class.  I expect all group members to contribute to the project and to be able to discuss all aspects of their groups work in detail. I reserve the right to reach out to any individual or group to ask for details on a given project after its submission and may incorporate this into grading of the final project. | |
|  |  | |
| **Homework:** | Homework will be assigned weekly and will involve coding. Each Homework will be due before the start of class on the specified due date.  You are required to submit any code that you write as part of your Homework submission. All code that is submitted should be easily re-runnable by me and the TA’s, and all submitted results should be easily reproducible.  Your grade for each assignment may reflect not only the solutions to the problems but also the quality and efficiency of the code that you write. It is recommended that you spend some time on each assignment making sure that the code you write follows the principles that we discuss in class and is scalable.  Homework should be completed individually. You may discuss assignments with other students but your write-ups and any supporting code must be unique and written by you. Any shared write-ups or shared code that we identify will be considered cheating. | |
|  |  | |
| **Cheating policy:** | This course will have a zero-tolerance cheating policy.  Any cheating incidence will be reported to the University, and may result in course failure or suspension from the University. | |