IEOR E4729: Advanced Programming for Financial Engineering (Fall 2013)
Syllabus and Course Logistics

Course Instructors:
Martin Haugh
332 S.W. Mudd Building
mh2078@columbia.edu
www.columbia.edu/~mh2078/

Fadi Kaddoura
314 S.W. Mudd Building
fadi@mikfs.com

TA’s: Andrew Ahn <aja2133@columbia.edu> and Hal Cooper <hc2683@columbia.edu>.

Course Website: All material will be posted on Columbia CourseWorks.

Class Time and Location: Tuesdays and Thursdays 5:40pm to 6:55pm in a classroom to be decided. Students should arrive on time and the use of cell-phones and laptops will not be permitted except for running specific course-related applications. Students may be cold-called regularly to answer questions in class.

About: This course is part of the programming track for the graduate students in Financial Engineering program and only suitably qualified graduate students in FE and OR students will be admitted. (Preference will be given to FE students in the event that the course is over-subscribed.) Python 2.7 will be the “default” programming language for the course but we will also use SQL, Excel-VBA, Bloomberg and the Bloomberg API’s. We will use Microsoft’s SQL Server Express as our database “server”. Details on downloading the various software components and configuring the programming environment will be provided in the first lecture.

Students taking this class should already know how to code and be familiar with object-oriented programming. The course will therefore focus as much on applications and the core project as it will on programming. These applications will draw mainly from finance but we will also consider some simple applications from “data-science” if time permits.

Core Project: There will be a core project that we will work towards throughout the course. The main goal of this project will be to programmatically calculate, store and produce reports on delta-adjusted and beta-adjusted risk exposures for a portfolio of equities and equity derivatives. To do this we will need to:

1. Design a (mini) security-master database to store issuers, underlying securities, industry sectors, contract sizes, etc.
2. Design a (mini) position time-series database to store position details, position size etc.
3. Use the Bloomberg API to get static data (issuer, underlying security etc.) and pricing data which we will then store in a database.
4. Produce formatted Excel spreadsheets displaying risk exposures by security, issuer, sector etc. These risk exposures will be obtained via pivot-tables and VBA.

**Prerequisites:** As stated above, students taking this class should already know how to code and be familiar with object-oriented programming. Students can demonstrate that they have the appropriate background by either: (i) passing the C++ refresher course that the IEOR department offers in the summer session of the FE program or (ii) demonstrating their knowledge via an interview with the course instructors. Students are not expected to know Python, SQL, Excel-VBA or be familiar with Bloomberg in order to take this class.

**Textbooks and Course Material:** We do not intend to follow any textbook as there is a lot of great material available online and we will certainly use and refer to some of this material as the course progresses. For example, there is a very nice Python tutorial available at

https://developers.google.com/edu/python/

We will be using a number of Python modules that are very useful for modeling and numerical work. They include the NumPy, SciPy, Matplotlib and possibly Pandas modules. Information and short tutorials on these modules can be found at

http://www.numpy.org/
http://www.scipy.org/
http://matplotlib.org/
http://pandas.pydata.org/

We will provide similar references for SQL, VBA etc. once the course begins. We will provide lecture slides and if necessary, we will also provide lecture notes describing the mathematics and modeling assumptions underlying the applications.

**Assignments:**
There will be approximately 10 assignments. In general students will be allowed to work in pairs but the policy may vary from assignment to assignment. Late assignments will **not** be accepted.

**Exams**
The course will have both a midterm and final exam. Any student who is unable to take an exam must have a very good reason for doing so, e.g., a medical emergency. Such students will take a makeup exam that will be more difficult than the regular exam. They will also need to obtain approval from the Dean’s office to take such an exam. Exam regrades may be requested by:

1. Explaining in a written statement why you think you should obtain additional points.
2. Submitting this statement and the exam to either the TA or one of the course instructors no later than one week after the exam was returned to the class.
means that if you failed to collect your exam within a week of it being returned to
the class, then you cannot request a regrade!)

It should be kept in mind that when a regrade is requested the entire exam will be
regraded and it is possible that your overall score could go down as well as up. **We
will also photocopy a subset of the exams before returning them to the class.**
This is intended to deter the very few people (hopefully there are no such people in this
class!) who might be tempted to rewrite parts of their exams before requesting a regrade.

**Grading**

A *tentative* grading scheme is: Assignments 35%, Midterm 30%, Final 35% but we re-
serve the right to deviate from this scheme if necessary.