IEOR E4728: Advanced Programming for Financial Engineering (Fall 2014)
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: To be confirmed.

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 MS in Financial Engineering (FE) program and only suitably qualified MSFE and MSOR students will be admitted. (In the event the course is over-subscribed preference will be given to **suitably qualified** first year MSFE students, then **suitably qualified** second year MSFE students and then **suitably qualified** MSOR students. We do expect the course to be oversubscribed and it is very unlikely that any places will be available for MSOR students.)

`Python 2.7` will be the "default" programming language for the course but we will also use a lot of `SQL` for the database component of the course. `Bloomberg` and the `Bloomberg` API will also be covered and, if time permits, we may also use `Excel-VBA` on occasion. 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.

We may also use and introduce `Github` as a version control system for our (and your!) software. A decision on this will not be made until the first week of class.

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" and operations research in general 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`, or alternatively through the *Pandas* `Python` module.

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 demonstrating their knowledge via an interview with the course instructors or convincing Jenny Mak. **Note:** Even if you convince us that you have an appropriate background there is no guarantee that you will b admitted to the class as places are limited. 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 and a textbook is not required. One very nice book, however, is "**Python for Data Analysis**" by Wes McKinney and most of the `Python` we'll need (except for the very important object-oriented aspects) is covered there. There is a lot of great Python and `SQL` 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/`

and a nice `SQL` tutorial is available at

`http://www.sqlcourse.com/index.html`

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 additional references as well as references for `SQL`, `Bloomberg` 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 expected to

work in pairs but the policy may vary from assignment to assignment. Late assignments will **not** be accepted.

Exams
The course will have **2** or **3** midterm exams and a final exam. The exams will be **multiple-choice** with **negative marking** for incorrect answers. 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. (This 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!)

Note however that with a multiple-choice exam there is very little room for debate!
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%, Midterms 35%, Final 30% but we reserve the right to deviate from this scheme if necessary.