THE UNIVERSITY OF HONG KONG
FACULTY OF BUSINESS AND ECONOMICS

FINA2390 Financial Programming and Databases

GENERAL INFORMATION

Instructor: Zigan Wang

Email: wangzg@hku.hk
Office: KK902
Phone: 2859 1045
Consultation times: By appointment

Pre-requisites: Prior programming experience is recommended but not required. I will provide a Coursera badge or Codecademy or assignment screening students unsure of their own background.
Co-requisites:
Mutually exclusive:

Course Website:
Other important details:

COURSE DESCRIPTION

This course provides undergraduate students a foundation in managing and analyzing financial datasets. Data analysis is a very important skill for the students to master. The first part of the course focuses on building skills – data manipulation using programming languages. The second part introduces various financial databases. Through practice on real-world financial datasets, students will learn methods used to warehouse and retrieve data for statistical computing. The course then turns to analytical methods with a focus on demonstrating these methods on data from various contexts in finance. Methods covered include manipulation of time series and panel data, statistical modeling and inference, simple textual analysis, classification and alternative datasets, etc. Problem sets and projects will be the primary mode of learning. Course learning will be supplemented with exposure to industry speakers from the local financial industry.

COURSE OBJECTIVES

1. Develop skills in database design, management, and access as would be expected of a first-year investment analyst.

2. Gain proficiency in programming and performing basic data cleaning, custodianship and data manipulation.

3. Gain a working understanding of different analytical methods used in finance and where the methods would be appropriate.

4. Gain fluency for at least one analytical method of the student’s choosing through course projects.

FACULTY LEARNING GOALS (FLGs)

FLG1: Acquisition and internalization of knowledge of the programme discipline
FLG2: Application and integration of knowledge
FLG3: Inculcating professionalism
FLG4: Developing global outlook
FLG5: Mastering communication skills
FLG6: Cultivating leadership

COURSE LEARNING OUTCOMES

Course Learning Outcomes	**Aligned Faculty Learning Goals (FLGs)**
CLO1 – Students will learn to store and access data efficiently using modern database storage methods, for example, loading input data and exporting output data.	CLO 1 – FLG 1-2

CLO2 – Students will gain an overview of analytical methods used in finance and their typical application, and demonstrate understanding of how to apply the methods through highly-supervised programming assignments.	CLO 2 – FLG 1,2,3,5
CLO3 – Demonstrate strong fluency in one analytical method of their own choice through course projects by presenting codes completely written by themselves.	CLO 3 – FLG 1,2,3,6
CLO4 – Creatively apply methods or data to solve specific industry problems, for example, using functions or packages that are not taught in the class.	CLO 4 – FLG 3,6
CLO5 – Communicate ideas with the instructor, tutor, and peer classmates.	CLO 5 – FLG 5

COURSE TEACHING AND LEARNING ACTIVITIES

Course Teaching and Learning Activities	**Expected Study Hours**	**Study Load (% of study)**
T&L1. - Lectures	16	30%
T&L2. – Problem sets.	35	70%
Total		100%

Assessment Methods	**Brief Description (Optional)**	**Weight**	**Aligned Course Learning Outcomes**
A1. Completion of 6-7 projects	Students will have multiple projects involving programming problems. Each step will enforce a particular skill. Students will be then given a choice of an open-ended problem using example data and be asked to solve the problem. Some projects are individual-based and some allow teamwork. The projects will be graded based on the effectiveness (whether correct output can be generated) and the algorithm (whether the computation is fast).	50%	CLO 1-4
A2. Presentation of two projects (one programming, one data) to test the understanding		30%	CLO 1-4
A3. General engagement and participation		20%	CLO 5
	Total	100%	

STANDARDS FOR ASSESSMENT

Course Grade Descriptors

A+, A, A-	The student would can competitively apply analytical methods used in the course independently.
B+, B, B-	The student appears to be able to apply analytical methods, but requires guidance.
C+, C, C-	The student has a conceptual understanding of methods applied in the course, but could not be expected to apply all methods used in the course.
D+, D	The student has shown effort, but a limited understanding of course content.
F	The student has not demonstrated effort to understand course content.

Assessment Rubrics for Each Assessment (Please provide us the details in a separate file if the space here is not enough)

For projects, the main driver of assessment will be accuracy with respect to the answers on which the assignments are based. An “A” quality course assignment looks professional and any discrepancies can be explained carefully. A “B” course assignment is mostly correct, but gaps in understanding remain. A “C” course assignment shows obvious gaps in understanding.

For the final course project, assessment will be based on quality of execution and originality of the investment idea. An A course project will demonstrate thorough understanding of course methods, careful consideration of pitfalls to analysis, and some element of originality. The work will be well communicated and easy to understand.

COURSE CONTENT AND TENTATIVE TEACHING SCHEDULE

I assume a twelve-week course schedule. I am going to provide assignments that are likely too many for a course of this length, but the number of assignments will be optional according to the course policy described in the section on **Course Policy**.

I also plan to have perhaps two or three course speakers. I know some Hong Kong-area financial professionals and I will see about their availability – attendance and a short write-up will be equivalent to an assignment.

Programming will be kept on Python.

1. **Week 1:** Introduction of the course, Python distributions, IDE and I/O
2. **Week 2:** Immutable and mutable data structures: string, list, dictionary, tuple, set
3. **Week 3:** Loop, exception handling
4. **Week 4:** Functions: lambda, apply, filter, reduce, map.
5. **Week 5:** Introduction of packages: numpy.
6. **Week 6:** Pandas and dataframe.
7. **Week 7:** OOP introduction, class, geolocation, multiprocessing
8. **Week 8:** Web scraping (beautifulsoup + requests, selenium)
9. **Week 9:** Introduction to WRDS financial database, Compustat, CUSIP/GVKEY
10. **Week 10:** Statistical analysis, causality inference
11. **Week 11:** Presentation of coding
12. **Week 12:** Presentation of coding
    a. Deadline of all projects

REQUIRED/RECOMMENDED READINGS & ONLINE MATERIALS (e.g. journals, textbooks, website addresses etc.)

https://www.coursera.org/
https://www.codecademy.com/

MEANS/PROCESSES FOR STUDENT FEEDBACK ON COURSE

Online response via Moodle site

COURSE POLICY (e.g. plagiarism, academic honesty, attendance, etc.)

The code of ethics and attendance policy will be applied.

To encourage participation and to discourage students from overinvesting in skills of low personal interest, I will have a relatively flexible policy regarding assignments:

- I will have 10 assignments, but students only have to do 7.
- The assignments will have extra credit.
- Students can work in groups, but a group can only have a maximum of 2 people.

ADDITIONAL COURSE INFORMATION (e.g. e-learning platforms & materials, penalty for late assignments, etc.)

I may prescribe self-learning materials to supplement students’ learning. Most of this will be optional. Late assignment policy will be to accommodate students who provide reasonable notice about competing obligations. Given that some assignments will be optional, students may also simply do other assignments given later in the course. I want to encourage self-motivated students to outperform, but do not want to discourage students excessively for a new module.