ISOM3400: Business Applications Development in Python

Summer 2024

Class Schedule*: Every Monday, Wednesday and Friday from 15 July to 05 August 2024,*
02:00pm~05:20pm Lecture, 05:30pm~06:20pm Lab
*August 7th (Wed), 02:00pm~****05:50pm*** *Lecture, 06:00pm~****06:50pm*** *Lab (in total 11 days)*
Venue*: Rm 1027 (LSK Bldg)*

Instructor:	Wanci YUAN, PhD candidate (wyuanab@connect.ust.hk)
Tel:	65860983
Office:	LSK 4063
Office Hour:	Thursday 02:00pm~06:00pm, by appointment

Teaching Assistant:	TBA
Tel:	TBA
Office:	TBA
Office Hour:	TBA

COURSE OVERVIEW

Python has recently become the most popular general-purpose programming language according to many polls among programmers. The scripting nature of Python allows fast development and easy maintenance of business applications. More importantly, the unparalleled community support makes Python increasingly powerful.

In this course, students will learn Python programming language in the context of business applications development. Business applications involve both business requirements and user requirements. Therefore, developers and programmers who design and develop business applications for organizations are required to meet those requirements. We believe that the combination of Python programming skills and business applications development will provide high practical value to students majoring in Information Systems as well as other related fields.

LEARNING OUTCOMES

By the end of this course, students will be able to:

(1) Acquire general programming knowledge with Python language

(2) Able to process data with Python language

(3) Able to design and develop business applications with Python language

(4) Conduct programming with team members effectively

TEACHING APPROACH

In general, the teaching approach of this course is based on the notion of sustained, deep learning by applying knowledge through programming, hands-on practices, and assignments.

Teaching & Learning Activities	Roles in the Course	Learning Outcomes addressed
Lecture	Explain key concepts to students using an active learning approach, in-class exercise, and after-class discussion of questions.	1,2,3
Laboratory	Apply concepts presented in lectures to hands- on exercises.	1,2,3
Assignment	It requires students to apply their knowledge and understanding in programming to solve business analytics problems.	2,3,4

EVALUATION

Components	**Percentage of grade**
A. Class/Lab Participation	10%
B. In-class Exercises	25%
C. Assignments	30%
D. Final Exam	35%
TOTAL:	100%

A. Class/Lab Participation (10%)

The fundamental programming concepts and associated practical examples will be covered during the lecture sessions. You are expected to attend lectures and contribute to class activities.

Additionally, participating in lab sessions is crucial to your learning, as it provides hands-on experience with more programming exercises. You are expected to attend lab sessions and submit assigned tasks during the lab time.

B. In-class Exercises (25%)

There are about **FIVE** in-class exercises throughout the semester. They will give you hands-on practice in Python programming in a setting where you can ask questions and collaborate with fellow students. Students' answers will be collected and graded. All scores will count towards the final grade. For each in-class exercise, the deadline of submission will **be the midnight on the**

same day the in-class exercise questions are given (23:59pm). There will be **NO makeup** in-class exercises for whatever reasons.

C. Assignments (30%)

There are TWO assignments. Students are expected to apply Python programming skills to solve practical business application problems.

Assignment 1 (individual) - (10%) This is an individual assignment. Each student needs to submit his/her program by the deadline. The detailed grading criteria will be stated clearly in the assignment document.

Assignment 2 (group) - (20%) This is a group assignment. A pre-assigned group is required to design and develop a business application. Details of the group project will be provided later in the semester.

D. Final Exam (35%)

There is a paper-based final exam (open note), which covers ALL topics taught in the semester. Details of the exam will be provided later in the semester.

Peer evaluation

Peer evaluation will be conducted for the group assignment. Students should make sure they make a fair contribution. We reserve the right to give less or even no credit to students who contribute significantly less or make no contributions.

Late policy

Turn in your work early if there is any uncertainty about your ability to turn it in at the due time. Submissions up to 24 hours late will have their grade reduced by 25%. **They will not be accepted for credit after 24 hours.**

Make-up policy

There will be no make-up exams except due to extraordinary circumstances beyond your control such as medical emergencies. Students have to submit appropriate documentation issued by a registered medical practitioner in order to be considered for a make-up exam.

Grade appeal

All scores will be uploaded to Canvas when ready. It is the student’s responsibility to check the scores and make sure they are correct. Any appeal to score must be filed through email to Wanci YUAN wyuanab@ust.hk within 72 hours after its release.

MATERIALS

1. MAIN READING

There is no required textbook. Lecture notes and extra exercises (either in pdf or ipynb format) will be posted on Canvas.

2. SUPPLEMENTAL READING

Many useful resources are also available online, for example, an online book Object-Oriented Programming in Python (https://python-textbok.readthedocs.io); Python Data Science Handbook by Jake VanderPlas (https://jakevdp.github.io/PythonDataScienceHandbook/); An Introduction to Statistical Learning, by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (Winner of the 2014 Eric Ziegel award from Technometrics). (https://www.statlearning.com/)

3. SOFTWARE

- Anaconda Navigator
    - Jupyter notebook
    - Python 3.7+
- Google Collaboratory
- Visual Studio Code (VS code)

4. *COURSE WEBSITE*

https://canvas.ust.hk. All course materials and announcements will be posted on this site. You are advised to check it regularly throughout the course.

OTHERS

Learning Environment

I welcome feedbacks on my teaching throughout the semester. You are encouraged to contact me at any time when you have any questions, suggestions, or concerns, or would like to ask for advice.

Academic Integrity

Academic integrity is a critical value of the university community. Integrity violations destroy the fabric of a learning community and the spirit of inquiry that is vital to the effectiveness of the University. Please remember the current university rule: "If a student is discovered cheating however minor the offence, the course grade will appear on the student's record with an X, to show that the grade resulted from cheating. This X grade stays on the record until graduation. If the student cheats again and ‘earns’ another X grade, the student will be dismissed from the University." Plagiarism is copying anything (text or ideas) from another source without citing that source. If you use another person's idea you must cite it, even if you rewrite the idea in your own words. Extreme care must be taken to avoid passing of other's work as one's own. You are required to provide appropriate citations when you use ideas and arguments or otherwise draw on others' work(e.g. if you use codes from Internet or your classmates). If you use research from another source or from the Web you MUST cite the source. This is true even if you use only the general idea and not the exact words.

What constitutes plagiarism in a coding class?

The course collaboration policy allows you to discuss the problems with other students but requires that you complete the work on your own. Every line of text and line of code that you submit must be written by you personally. You may not refer to another student's code, or a "common set of code" while writing your own code. You may, of course, copy/modify lines of code that you saw in lecture or lab.

Tentative Lecture Schedule

Date	Topics	Assignments/Due Dates
July 15	Course Introduction Python Basics: Data, Data types, and Operators	
July 17	Data Structures: Lists, Tuples, Dictionaries, and Sets	Asg.1 Release
July 19	Control Structures: if, for, while, and try statements	
July 22	Functions and Classes I	
July 24	Functions and Classes II	
July 26	Web Scraping with *Beautiful Soup*	Asg. 2 Release Asg.1 Due
July 29	Web Automation with *Selenium*	
July 31	Data Analysis and Manipulation with *pandas*	
August 2	Web App Development with *Dash*	
August 5	Visualization with *matplotlib*; Working with text data	
August 7	Final Exam	