ISOM 3400 - Business Applications Development in Python

Spring 2025

Instructor: Jing WANG, Ph.D.
Email: jwang@ust.hk Begin subject: [ISOM3400]
Tel: 3469-2125
Office: LSK 4044
Office Hours: By appointment

Teaching Assistant: Aaron LEE
Email: imaaron@ust.hk Begin subject: [ISOM3400]
Tel: 2358-7638
Office: LSK 4065
Office Hours: By appointment

Teaching Assistant: Cetus WONG
Email: imhtwong@ust.hk Begin subject: [ISOM3400]
Tel: 2358-5728
Office: LSK 4048
Office Hours: By appointment

Class Schedule

Lecture

Section	Date	Time	Venue
L2	Tu	4:30pm – 5:50pm	Rm 6573, Lift 29-30
L2	Th	4:30pm – 5:50pm	Rm 6573, Lift 29-30
L3	Tu	9:00am-10:20am	Rm 2404, Lift 17-18
L3	Th	9:00am-10:20am	Rm 2404, Lift 17-18

Lab

Section	Date	Time	Venue
LA1	Tu	4:30pm - 5:20pm	G021 (LSK Bldg)
LA2	Tu	10:30am – 11:20am	G021 (LSK Bldg)
LA3	Mo	10:30am – 11:20am	G021 (LSK Bldg)
LA4	Mo	4:30pm – 5:20pm	G021 (LSK Bldg)

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.

COURSE GOALS

This course will provide students with skills and knowledge of Python programming and experience in designing and developing business applications.

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 in Python
 (4) Conduct programming with team members effectively

COURSE DESCRIPTION

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.

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

Components	Percentage
A. Labs	5%
B. Assignment 1	15%
C. Assignment 2	15%
D. Final Exam – Programming concepts	25%
E. Final Exam – Advanced techniques	20%
F. Final Exam – Business applications	20%
TOTAL:	**100%**

A. Labs

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 to solve your puzzles and hone your programming skills.

B. Assignments

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

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%; those up to 48 hours late will have their grade reduced by 50%. They will not be accepted for credit after two days.

Honor-code policy

The basic presumption is that the work you submit is your own. Every line of text and line of code that you submit must be written by you personally.

However, occasionally, it may be necessary to ask someone for help or use generative AI tools. You are permitted to do so, provided you meet the following two conditions:

1. You acknowledge any help received on the work you hand in. That is, you must include a comment in your homework submission that clearly states the name of the student, book, or online reference from which you received assistance.

2. You *understand* the work that you hand in, so that you could explain the reasoning behind the parts of the work on which you received assistance from others.

Submissions that fail to properly acknowledge help from other students or non-class sources **will receive no credit**. Copied work **will receive no credit.** Any and all violations **will be reported** to the University administration.

C.   Final Exam

There is an **open-note paper-based** final exam, which covers ALL topics taught in the semester. It is further divided into 3 parts as listed in the evaluation section. Details of the exam will be provided later in the semester.

Make-up policy

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

Grade appeal

All scores will be uploaded to Canvas when ready. It is always the student's responsibility to check the scores and make sure they are correct. Any appeal to score must be filed through email to jwang@ust.hk. No appeal to a particular score is allowed 72 hours after its release.

D.   In-class Exercises

There are 5 in-class exercise sessions scheduled throughout the semester. Some of the exercise questions are adapted from past exams. They will help you reinforce programming concepts in a timely manner and gain hands-on programming experience in a setting where you can ask questions and collaborate with fellow students. **No answers from these sessions will be collected for grading**.

MATERIALS
1.   **MAIN READING**

This course has no required textbook. Lecture notes and extra exercises (either in .pdf or .ipynb or .py format) will be posted on the course website.

2.   **SOFTWARE**
* Google Collaboratory
* Anaconda Navigator
  o Jupyter notebook
  o Python 3.10+
* Visual Studio Code (VS code)

Feedback

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

Date	Topics	Assignments/ Due dates
Feb 4	Course Introduction	
Feb 6	Python Basics: Data, Data types, and Operators	
Feb 11		
Feb 13	Data Structures: Lists, Tuples, Dictionaries, and Sets	Add/Drop deadline: Feb 15
Feb 18		
Feb 20	**In-class exercise session 1**	
Feb 25	Control Structures: if, for, while, and try statements	
Feb 27		
Mar 4		
Mar 6	**In-class exercise session 2**	Asg. 1 Release
Mar 11	Functions, Classes, and Modules	
Mar 13		
Mar 18		
Mar 20	**In-class exercise session 3**	
Mar 25	Web Scraping with *Beautiful Soup*	Asg. 1 Due
Mar 27		
Apr 1	*Midterm Break*	
Apr 3	*Midterm Break*	
Apr 8		
Apr 10	**In-class exercise session 4**	Asg. 2 Release
Apr 15	Data Analysis and Manipulation with *pandas*	
Apr 17	Web App Development with *Streamlit*	
Apr 22		
Apr 24		
Apr 29		Asg. 2 Due
May 1	*Public Holiday*	
May 6	**In-class exercise session 5**	
May 8	Final Revision	

TENTATIVE LAB SCHEDULE

Week/Date	Topics
Feb 3/4	Use of Colab, Setup of VS Code & Anaconda
Feb 10/11	Python Basics
Feb 17/18	Data Structures: List, Tuple
Feb 24/25	Data Structures: Dictionary, Set
Mar 3/4	Control Structures I
Mar 10/11	Control Structures II
Mar 17/18	Function and Class I
Mar 24/25	Web Scraping I: HTML and CSS Basics
Mar 31/Apr 1	*Midterm Break*
Apr 7/8	Web Scraping II: Beautiful Soup
Apr 14/15	Streamlit Setup
Apr 21/22	Web App Development I
Apr 28/29	Web App Development II

* Note both the lecture, lab, and assignment schedules are tentative and subject to change without notice.