Business Applications Development in Python
ISOM3400

3 Credits

Prof. James Kwok

jkwok@ust.hk

Office hours: By appointment

Course goals

This course provides students with foundational skills in Python programming and hands-on experience in designing and developing business applications.

Intended Learning Outcomes (ILOs)

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

1. **Attain Proficiency in Python Programming** - Develop a solid understanding of programming concepts using the Python language.
2. **Capability to Design and Develop Python-based Business Applications -** Acquire the skills necessary to design and construct business applications utilizing Python.
3. **Effective Collaborative Programming -** Cultivate the ability to collaborate efficiently with team members while engaging in programming tasks.

Course description

Python has become the leading general-purpose programming language, known for its simplicity, rapid development capabilities, and strong community support. This course introduces Python in the context of business application development, emphasizing the importance of aligning technical solutions with business and user requirements.

Students majoring in Information Systems and related fields will gain practical value from combining Python expertise with business application development skills. As a programming-focused course, students are expected to independently explore online resources and research Python syntax beyond the provided materials to complete assignments and projects.

Assessment and Grading

This course will be assessed using **criterion-referencing** and grades will not be assigned using a curve. Detailed rubrics for each assignment are provided below, outlining the criteria used for evaluation.

Summary Table

Assessment Task	**Contribution to Overall Course grade (%)**	**Due date**
A. In-class activity – Group Exercise	**10%**	As per course schedule
B. Assignment	**20%**	As per course schedule
C. Final Exam – Programming Logic	**20%**	To be announced
D. Final Exam – Advanced Techniques	**25%**	To be announced
E. Final Exam – Business Applications	**25%**	To be announced

A. In-class activities – Group Exercise (10%)

Students will complete **ONE** in-class activity during the semester. Please note that **no make-up opportunities** will be provided under any circumstances.

Groups may consist of **two to five members**. Each group is expected to apply their Python programming skills to solve a business-related problem using Visual Studio Code (VS Code) and Google Colab. At the end of the session, group submissions will be collected and assessed. All contributing members will receive the same score. However, individuals who do not participate meaningfully ("free riders") will not receive credit.

It is the **responsibility of group members** to report any instances of free riding, supported by appropriate evidence. Reports must be submitted **within five days** of the exercise.

Technical Requirements

Students are responsible for ensuring that VS Code, Google Colab, and all required Python packages are properly installed and functioning prior to the exercise.

B. Individual Assignment (20%)

This course includes an individual assignment requiring students to apply their Python programming skills to solve real-world business challenges. There is only one individual assignment. The specifics of the assignment will be communicated at a subsequent point in the course.

Selected students will showcase their findings during the final class session through the presentation of their video files.

C. Final Exam – Programming Logic (20%)

The final exam will comprehensively assess all topics covered throughout the semester. This section will focus on evaluating students' understanding of programming logic.

D. Final Exam – Advanced Techniques (25%)
This section of the final exam will test students on advanced programming techniques, drawing from the full range of topics discussed during the semester.

E. Final Exam – Business Applications (25%)
The final exam will also include a section dedicated to business applications, assessing students' ability to apply programming concepts in practical business contexts.

Late Submission Policy:
Late submissions within 3 hours will result in a 30% mark deduction. Submissions more than 3 hours late will not be accepted.

Feedback on all assessments:

- Feedback on all assignments and assessments will be provided within 10 working days.
- A summary highlighting common mistakes or key deficiencies in answering questions will be shared with students.
- Additionally, students can schedule a meeting with our Teaching Assistant (TA) to review their assignments and examination papers, gaining insights into their mistakes and deficiencies. This review session must take place within a specified deadline, typically two working days after the scores are released. After this deadline, students will not be allowed to review their assignment and examination papers.

Arrangements for the Make-up Final Exam

Make-up final exams will only be conducted in cases of exceptional circumstances beyond a student’s control, such as medical emergencies. If a student is absent due to a medical emergency, they must submit relevant documentation from a registered medical practitioner to the course instructor via email. This documentation is required for consideration for a make-up exam. The make-up exam will be in essay format, and the maximum score a student can achieve is **50%** of the total score of the final exam.

(**Attention**: Students who are eligible to take the make-up exam are required to compose a research article consisting of an introduction, references, proper citations, and other essential sections. This article must be completed within a few hours of its assignment. Please note that there will be **no opportunity for a second make-up exam** under any circumstances. Failing to submit the research article for any reason, such as email or internet issues, will result in a grade of ZERO for the exam.**)**

Score Posting and Appeal Policy

Upon completion, all scores will be posted on Canvas. It is incumbent upon the student to review their scores and verify their accuracy. If any discrepancies arise, score appeals must be submitted via email to jkwok@ust.hk. It is important to note that score appeals will not be entertained once the designated checking/appeal period has elapsed (e.g., two working days after the score release) if applicable.

[*If a student is unable to review their paper within the designated checking period, the score will be considered final by default. Unfortunately, we will not be able to modify or correct the score after the checking or appeal period has ended. All scores will be finalized 10 working days after the appeal deadline, except for Final Exam scores. Once scores are finalized, no further changes can be made.*]

Final Exam Paper Checking

Upon completion of the grading process for the Final Examination papers, the corresponding scores and detailed marking will be uploaded to Canvas. Students are required to review and verify their scores promptly. In compliance with the University's submission deadlines for final grades, **a two-hour paper checking session** will be organized and facilitated by the Teaching Assistant. During this session, students may inspect the marking of their examination papers in person. Following the conclusion of the paper checking session, all assessment scores will be deemed final, and no subsequent modifications or adjustments will be permitted under any circumstances.

Mapping of Course ILOs to Assessment Tasks

Assessment Task	Mapped ILOs	Explanation
A. In-class activities	ILO1, ILO2, ILO3	This task evaluates students' ability to apply Python programming skills (ILO1), collaboratively design and develop business solutions (ILO2), and work effectively in teams during programming tasks (ILO3). Students demonstrate their understanding of business problems and apply Python tools in a group setting using VS Code and Google Colab.
B. Assignment	ILO1, ILO2	This individual task measures students' proficiency in Python programming (ILO1) and their ability to design and implement business applications independently (ILO2). It requires students to solve real-world business challenges using Python, showcasing their technical and analytical skills.
C. Final Exam – programming logic	ILO1	This section assesses students' understanding of programming logic and core Python concepts (ILO1). Students are expected to analyze code structures, predict outputs, and identify logical errors.
D. Final Exam – advanced techniques	ILO1, ILO2	This section evaluates students' ability to apply advanced Python programming techniques (ILO1) and construct

		complex business applications (ILO2). It tests their depth of knowledge and problem-solving capabilities.
E. Final Exam – business application	ILO2, ILO3	This section assesses students' ability to align programming solutions with business needs (ILO2) and demonstrate collaborative and practical application of Python in business contexts (ILO3). It emphasizes real-world relevance and teamwork in business application development.

Final Grade Descriptors:

Grades	Short Description	Elaboration on subject grading description
A	Excellent Performance	Demonstrates exceptional proficiency in Python programming, consistently applying advanced techniques to solve complex business problems. Shows a deep understanding of programming logic and business application design. Collaborates effectively in team settings, contributing meaningfully to group tasks and demonstrating leadership in problem-solving.
B	Good Performance	Shows solid understanding of Python programming and applies it effectively to business challenges. Demonstrates good grasp of programming logic and can design functional business applications. Participates actively in group work and contributes to collaborative programming tasks.
C	Satisfactory Performance	Possesses basic understanding of Python programming and can apply it to solve standard business problems. Demonstrates adequate understanding of programming logic and business application design, though may require assistance. Participates in group tasks with moderate contribution.
D	Marginal Pass	Has limited understanding of Python programming and struggles to apply it effectively to business problems. Demonstrates weak grasp of programming logic and business application design. Contribution to group tasks is minimal and inconsistent.
F	Fail	Demonstrates insufficient understanding of Python programming and fails to apply it to business challenges. Lacks comprehension of programming logic and business application design. Shows poor or no contribution to group tasks and collaborative efforts.

Student Rubrics

Use the following rubrics to guide you for the assessment tasks that you submit in this course.

Assessment: In-class Activities

Criteria: Application of Python Concepts
• **Excellent**: Demonstrates thorough understanding and applies Python concepts accurately and creatively. • **Good**: Applies Python concepts correctly with minor errors or omissions.

• **Satisfactory**: Applies basic Python concepts with noticeable errors or lack of depth. • **Marginal**: Struggles to apply Python concepts effectively. • **Fail**: Fails to apply Python concepts or demonstrates misunderstanding.
Criteria: Participation and Contribution
• **Excellent**: Actively participates in all activities, contributes insightful ideas, and collaborates effectively with peers. • **Good**: Participates regularly, contributes relevant ideas, and works well with peers. • **Satisfactory**: Participates occasionally, contributions are basic, and collaboration is minimal. • **Marginal**: Rarely participates, contributions lack relevance, and limited collaboration. • **Fail**: Does not participate or contribute meaningfully to activities.
Criteria: Meeting Stated Requirements
• **Excellent**: Fully meets all requirements with a robust and complete solution. • **Good**: Meets most requirements with minor omissions. • **Satisfactory**: Meets basic requirements but lacks completeness. • **Marginal**: Partially meets requirements; key elements are missing. • **Fail**: Fails to meet the majority of requirements.

Assessment: Assignment

Criteria: Use of Python
• **Excellent**: Advanced use of Python with efficient, well-structured code and appropriate libraries. • **Good**: Correct use of Python with mostly efficient and readable code. • **Satisfactory**: Basic use of Python with some inefficiencies or structural issues. • **Marginal**: Limited use of Python with significant issues in code structure or logic. • **Fail**: Incorrect or missing use of Python in the assignment.
Criteria: User Friendliness
• **Excellent**: Highly intuitive and user-friendly interface with clear instructions and smooth navigation. • **Good**: Mostly user-friendly with minor usability issues. • **Satisfactory**: Basic usability with some confusing elements or unclear instructions. • **Marginal**: Poor usability with significant navigation or instruction issues. • **Fail**: Unusable interface or lack of user guidance.
Criteria: Meeting Stated Requirements
• **Excellent**: Fully meets and exceeds all stated requirements with innovative enhancements. • **Good**: Meets all stated requirements with minor omissions. • **Satisfactory**: Meets most requirements but lacks completeness or clarity. • **Marginal**: Meets few requirements with major omissions. • **Fail**: Does not meet the stated requirements.
Criteria: Presentation
• **Excellent:** (a**)** Presentation is clear, engaging, and professional with excellent video and audio quality. (b) Demonstrates thorough preparation and understanding of the material. • **Good:** (a) Presentation is clear but may have slight issues with video or audio quality. (b) Covers most key points effectively, but minor details could be enhanced. • **Satisfactory:** (a**)** Presentation is understandable but lacks engagement or quality in video/audio. (b) Some key points are covered, but it feels incomplete or rushed.

• **Marginal:** (a**)** Presentation is unclear or poorly organized, with major issues in video or audio quality. (b) Fails to engage the audience or misses significant content. • **Fail:** (a) Presentation is unclear or unprofessional, with very poor video/audio quality. (b) Lacks adequate coverage of the topic, showing little effort or preparation.

Assessment: Final Exam – Programming Logic

Criteria: Detecting and Correcting Errors
• **Excellent**: Accurately detects and corrects all errors in the code, demonstrating a clear understanding of program logic and effective troubleshooting. • **Good**: Detects and corrects most errors but misses a few minor issues or overlooks some details. • **Satisfactory**: Identifies and corrects some errors, though several may remain, affecting the overall functionality of the program. • **Marginal**: Detects and corrects only a few errors, with many issues still affecting the program's execution. • **Fail**: Fails to detect or correct most errors, resulting in a program that does not function as intended.
Criteria: Predicting Outputs
• **Excellent**: Accurately predicts the outputs for all test cases, providing detailed reasoning behind predictions and demonstrating deep understanding of program flow. • **Good**: Predicts the outputs for most test cases, with minor errors in reasoning or predictions. • **Satisfactory**: Provides reasonable output predictions but may misinterpret some program behaviour or overlook edge cases. • **Marginal**: Predicts outputs inaccurately, with significant errors in reasoning or a misunderstanding of program logic. • **Fail**: Fails to predict outputs correctly, with no understanding of the underlying program flow.
Criteria: Revising and/or Enhancing Code
• **Excellent**: Makes insightful revisions and enhancements to improve code efficiency, readability, and functionality, adhering to best practices in programming. • **Good**: Makes adequate revisions and enhancements, but some areas of the code could be further optimized or improved. • **Satisfactory**: Makes basic revisions or enhancements, but the code still lacks efficiency or clarity in some parts. • **Marginal**: Makes minimal revisions or enhancements, with no noticeable improvement in code quality. • **Fail**: Fails to make any meaningful revisions or enhancements to the code, leaving it inefficient or non-functional.

Assessment: Final Exam – Advanced Techniques

Criteria: Application of Advanced Programming Techniques
• **Excellent**: Applies advanced techniques such as OOP effectively. • **Good**: Uses advanced techniques with minor implementation issues. • **Satisfactory**: Basic use of advanced techniques with limited understanding. • **Marginal**: Attempts advanced techniques with poor execution.

• **Fail**: Fails to apply advanced programming techniques.
Criteria: Effective Use of Programming Practices
• **Excellent**: Demonstrates best practices including modularity, documentation, and testing. • **Good**: Applies good practices with minor omissions. • **Satisfactory**: Applies basic practices with noticeable gaps. • **Marginal**: Limited use of programming practices. • **Fail**: Poor or no use of programming practices.
Criteria: Meeting Stated Requirements
• **Excellent**: Fully meets and enhances all requirements with advanced features. • **Good**: Meets all requirements with minor gaps. • **Satisfactory**: Meets most requirements with basic implementation. • **Marginal**: Meets few requirements with major issues. • **Fail**: Does not meet the stated requirements.

Assessment: Final Exam – Business Applications

Criteria: Solution Development and User Requirements
• **Excellent**: Develops comprehensive solutions aligned with user needs and business context. • **Good**: Develops appropriate solutions with minor gaps in user alignment. • **Satisfactory**: Basic solution development with limited user consideration. • **Marginal**: Incomplete solutions with poor user alignment. • **Fail**: Fails to develop a usable solution.
Criteria: Application of Programming Techniques
• **Excellent**: Effectively applies programming techniques to solve business problems. • **Good**: Applies techniques with minor inefficiencies. • **Satisfactory**: Basic application with limited problem-solving effectiveness. • **Marginal**: Poor application with minimal relevance. • **Fail**: Fails to apply programming techniques appropriately.
Criteria: Effective Use of the Programming Language
• **Excellent**: Demonstrates mastery of Python with efficient and readable code. • **Good**: Uses Python effectively with minor issues. • **Satisfactory**: Basic use of Python with noticeable inefficiencies. • **Marginal**: Limited use of Python with major issues. • **Fail**: Fails to use Python effectively.
Criteria: Meeting Stated Requirements
• **Excellent**: Fully meets and exceeds all business application requirements. • **Good**: Meets all requirements with minor gaps. • **Satisfactory**: Meets most requirements with basic implementation. • **Marginal**: Meets few requirements with major issues. • **Fail**: Does not meet the stated requirements.

Efficient Email Communication Guidelines

To ensure prompt assistance, please include [**Course Code** - L**X**] (**X** being the section number), e.g., [**ISOM3400**-L**1**] at the start of your email's subject line. Neglecting this may lead to delays in our response time.

Anticipate a surge in email volume as deadlines approach. For timely support, address your queries ahead of time and utilize instructor and TA office hours.

Kindly note that **direct assignment answers won't be furnished by the instructor or TAs.** Your understanding and collaboration are appreciated.

Policy on the Use of Generative AI

Students are permitted to utilize generative artificial intelligence (AI) tools exclusively for enhancing programming tasks within this course. Nonetheless, students are obligated to duly acknowledge and credit any employment of generative AI. In the context of producing **video presentations, employing generative AI tools is strictly prohibited** for students.

- Leveraging Gen AI tools, individuals can effortlessly generate content devoid of grammatical errors. As a result, during assessment, we presuppose that the content is devoid of any grammatical blunders.
- We anticipate students to acquire coding skills by independently employing Gen AI tools. For instance, when seeking additional practice and examples, Gen AI tools can provide valuable assistance.

	Gen AI Tools
Group Exercise	✓
Assignment	✓
Final Exam	✗
Lecture and Lab	✓
Outside the class (for learning)	✓

Student learning resources

Text and Reference Books

No particular textbooks or reference books are mandatory for this course. The learning materials will comprise diverse readings accessible on Canvas.

Course Website
Course content updates and other pertinent information will be communicated through the course website - http://canvas.ust.hk. It is advisable for students to consistently monitor this platform throughout the semester.

Software Requirements

- Python 3.13+
- Google Colaboratory
- Visual Studio Code (VS Code)
- Gen AI, e.g., Copilot

Course schedule

The course is offered in lecture sessions and laboratory sessions.

L1:	Monday	16:30 – 17:50	LSK 1001
	Friday	12:00 – 13:20	LSK 1001
LA1:	Tuesday	13:30 – 14:20	LSK G021

Tentative Course Schedule. Please visit Canvas for updated schedule, readings, and assignments.

Schedule of Lecture (Tentative)

Week	Date	Topics	Assignment Release/Due
1	2 Feb	Intro. to Course and Programming Introduction to Python and Business Applications, Gen AI	
	6 Feb	Python Basics: Variables, Data Types, Operators, Inputs and Outputs	
2	9 Feb	Data structure: Lists, Tuples, and Dictionaries	
	13 Feb	Data Validation: If-else, for, while, try-except	Add/Drop deadline: **Feb 14th**
3	16 Feb	Data Validation: If-else, for, while, try-except	
	20 Feb	Examples of Business Application	
4	23 Feb	Practice	
	27 Feb	Functions and Classes	
5	2 Mar	Functions and Classes	
	6 Mar	Functions and Classes	
6	9 Mar	Functions and Classes	
	13 Mar	Functions and Classes	
7	16 Mar	Practice	
	20 Mar	**In-class activity - Group Exercise**	
8	23 Mar	Web Automation – Selenium	
	27 Mar	Web Automation – Selenium	
9	30 Mar	Web Automation – Selenium	**Assignment**: Release on **Mar 30th**
	3 Apr	**No Class**	
	6 Apr	**No Class**	
	10 Apr	Web Automation - Selenium	
10	13 Apr	Web Automation - Selenium	
	17 Apr	Web Applications Development – Streamlit	

11	20 Apr	Web Applications Development – Streamlit	
	24 Apr	Web Applications Development – Streamlit	
12	27 Apr	Web Applications Development – Streamlit	
	1 May	**No Class**	**Assignment**: Due on **May 3rd**
13	4 May	Web Applications Development – Streamlit	
	8 May	**Presentation and Revision**	

Schedule of Laboratory (Tentative)

Week	Date	No.	Topics
1	2, 3 Feb	LA01	VSCode & Anaconda, Gen AI, Virtual Env., data, data types
2	9, 10 Feb	LA02	Lists, Tuples, and Dictionaries
3	16, 17 Feb	LA03	if-else, for, while, try-except
4	23, 24 Feb	LA04	Functions and Classes
5	2, 3, Mar	LA05	Functions and Classes
6	9, 10 Mar	LA06	Functions and Classes
7	16, 17 Mar	LA07	Functions and Classes
8	23, 24 Mar	LA08	Selenium
9	30, 31 Mar	LA09	Selenium
	6, 7 Apr		**No Lab**
10	13, 14 Apr	LA10	Web Applications Development
11	20, 21 Apr	LA11	Web Applications Development
12	27, 28 Apr	LA12	Web Applications Development
13	4, 5 May	LA13	Web Applications Development

***Note**: If lab sessions are canceled due to public holidays or other reasons, a Zoom recording will be provided when available.

Contact Details for Instructor and TA

Prof. Kwok's office is located in room LSK4080, and he extends a warm invitation for you to visit during his office hours or at your convenience for any queries you may have. For urgent concerns, feel free to reach out via email (jkwok@ust.hk) or phone (2358-7652); however, he does emphasize that email is the preferred mode of communication as he frequently monitors it. Additionally, the Teaching Assistant (TA) assigned to this course is available to address inquiries related to grading, attendance, assignments, and any administrative matters.

Academic honesty

Upholding academic integrity stands as a fundamental principle within our university community. Any breach of integrity undermines the foundation of our learning environment and the essence of inquiry that is vital for the institution's effectiveness. I maintain a zero-tolerance stance towards cheating, and no exceptions will be entertained. Students found engaging in acts of cheating, plagiarism, or any form of academic dishonesty will face a reduction of their course grade by a minimum of one letter grade. Moreover, it is my responsibility to report any instances of unethical conduct or indications of dishonesty in this course to the University.

Please bear in mind the current university regulation: any occurrence of cheating, irrespective of its magnitude, will result in an "X" grade notation on the student's academic record, signifying that the grade was attained through dishonest means. This "X" grade will persist on the student's record until graduation. Should a student be caught cheating again and subsequently receive another "X" grade, they will be dismissed from the University.

Plagiarism encompasses the act of copying text or ideas from external sources without appropriate citation. Even if you rephrase the concept using your own words, citing the origin is necessary when utilizing someone else's idea. It is imperative to exercise extreme caution to prevent presenting someone else's work as your own. Proper citations are obligatory when incorporating external sources' ideas, arguments, or any content. Whether drawing from research or the Internet, it is mandatory to acknowledge the source, even if you employ the general notion rather than verbatim wording.

Learning environment

I wholeheartedly embrace feedback on my teaching during the entirety of the semester. I strongly encourage you to reach out to me or my TA whenever you have questions, suggestions, concerns, or if you seek advice. Your input is valued and will contribute to enhancing the learning experience. Feel free to contact us at your convenience.