Business Programming in VBA
ISOM3230

3 Credits

Prof. James Kwok
jkwok@ust.hk
Office hours: By appointment

Course goals

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

Intended Learning Outcomes (ILOs)

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

1. Programming Techniques & User Requirements – Apply fundamental and advanced programming principles to develop effective solutions for business challenges while addressing and implementing user needs.
2. Program Logic & Analysis – Assess and explain program structures, accurately predicting outputs and identifying potential issues.
3. Development & Debugging – Employ programming techniques to design, test, and refine programs, effectively detecting and resolving logical and runtime errors.
4. Effective Use of the Programming Language – Utilize the programming language efficiently and effectively, following common programming practices to ensure optimal program development and performance.

Course description

This course is tailored to equip students with a comprehensive understanding of programming, specifically focusing on programming for business applications in MS Excel. Through this course, students will delve into the rationale behind incorporating programming into their respective professions, the significance of constructing business applications, and the impact of these applications on business workflows. Additionally, they will explore the heightened utility and advantages that programming can bring to the realm of business applications.

Furthermore, students will become adept at grasping fundamental programming syntax and structure. They will gain the proficiency to construct rudimentary business applications utilizing high-level programming languages.

It is important to note that this course is centered around programming. Students are expected to engage with online resources independently to enhance their learning. Throughout the course, students will be tasked with researching VBA syntax through sources like Google, which might not be extensively covered in the course materials but are necessary for fulfilling course assignments and tasks.

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. Group Exercise	**10%**	As per course schedule
B. Assignment	**20%**	As per course schedule
C. Final Exam – Programming Logic	**20%**	NA
D. Final Exam – Advanced Techniques	**25%**	NA
E. Final Exam – Business Applications	**25%**	NA

A. Group Exercise – Group (10%)

Throughout the semester, there will be **ONE** group exercise. Please be advised that no makeup exercise will be provided under any circumstances.

Students may form groups of **two to five members** for this exercise. Groups are expected to apply their VBA programming skills and knowledge to address business challenges. At the conclusion of the class, group programs will be collected and evaluated. All members of the group, except those who did not contribute or did not contribute sufficiently (free riders), will receive the same score for the exercise.

It is the responsibility of group members to report any free riding activity along with evidence during the group exercise. Such cases must be reported within 5 days after the group exercise.

Important Reminder: Students are responsible for ensuring the compatibility and functionality of MS Excel, notebook computers, and other relevant software during group exercises.

B. Assignment – Individual (20%)

The aims of the assignments encompass the analysis and resolution of business predicaments through VBA utilization. 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%)

A comprehensive Final Exam will encompass **ALL topics** covered during the semester. This section focuses on questions related to programming logic.

D. Final Exam – Advanced Techniques (25%)

A comprehensive Final Exam will encompass **ALL topics** covered during the semester. This section focuses on questions related to advanced programming techniques.

E. Final Exam – Business Applications (25%)

A comprehensive Final Exam will encompass **ALL topics** covered during the semester. This section focuses on questions related to business applications.

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. Group Exercise	ILO1, ILO2, ILO3, ILO4	This task evaluates students' ability to apply programming techniques to solve business challenges (ILO 1), assess and explain program structures (ILO 2), develop and debug programs effectively (ILO 3), and utilize the programming language efficiently (ILO 4). Through collaborative exercises, students will demonstrate their understanding of user requirements, logic analysis, and debugging strategies while adhering to best programming practices.
B. Assignment	ILO1, ILO2, ILO3, ILO4	This task measures students' proficiency in designing and developing business applications (ILO 1), analyzing program logic to predict outputs and identify issues (ILO 2), debugging and refining programs (ILO 3), and writing efficient, well-structured code following best practices (ILO 4). The assignment requires students to create a functional program addressing a real-world business scenario, ensuring logical consistency and optimal performance.

C. Final Exam – Programming Logic	ILO1, ILO2, ILO3, ILO4	This exam assesses students' ability to analyze program structures and predict their outputs accurately (ILO 2). It requires them to demonstrate logical reasoning and debugging skills (ILO 3) by identifying and correcting logical errors in given code snippets. The exam also tests students' understanding of efficient programming practices (ILO 4) to ensure code clarity and maintainability.
D. Final Exam – Advanced Techniques	ILO1, ILO2, ILO3, ILO4	This task evaluates students' application of advanced programming principles in solving complex business problems (ILO 1), their ability to critically analyze intricate program structures (ILO 2), and their debugging efficiency in identifying runtime errors (ILO 3). Additionally, it measures students' ability to optimize code performance and adhere to industry-standard programming practices (ILO 4).
E. Final Exam – Business Application	ILO1, ILO2, ILO3, ILO4	This exam assesses students' ability to develop business applications that effectively address user requirements (ILO 1), logically structure programs to ensure functionality and maintainability (ILO 2), apply debugging techniques for robust program execution (ILO 3), and utilize programming languages efficiently for optimal performance in business contexts (ILO 4).

Final Grade Descriptors:

Grades	Short Description	Elaboration on subject grading description
A	Excellent Performance	Demonstrates an exceptional ability to apply advanced programming principles to solve complex business challenges, consistently addressing user requirements with a high level of precision. Shows a thorough understanding of program structures, accurately predicting outputs and identifying potential issues. Proactively employs effective debugging and development techniques, consistently refining programs and resolving errors with ease. Uses the programming language efficiently, adhering to best practices, ensuring optimal program performance and development.
B	Good Performance	Shows a solid understanding of programming principles and can effectively apply them to business challenges while addressing user needs. Demonstrates good understanding of program structures, predicting outputs and identifying issues with accuracy. Effectively employs debugging and development techniques to resolve common errors. Uses the programming language competently, following common practices to ensure solid program development and performance.
C	Satisfactory Performance	Possesses a basic understanding of programming principles and can apply them to solve business challenges, though may struggle to fully address all user requirements. Demonstrates adequate understanding of program structures but may miss some potential issues. Uses debugging techniques to resolve some errors, though

		occasionally requires additional assistance. Utilizes the programming language with a basic understanding of common practices, producing functional but not optimal programs.
D	Marginal Pass	Has a limited understanding of programming principles and is able to apply basic techniques to solve simple business challenges, but struggles to meet user requirements fully. Demonstrates a weak grasp of program structures, often failing to predict outputs or identify issues. Limited ability to detect or resolve errors in programs, requiring significant assistance. Uses the programming language at a basic level, but with inconsistent adherence to common programming practices, resulting in suboptimal program performance.
F	Fail	Demonstrates insufficient understanding of programming principles and is unable to apply them to solve business challenges. Lacks an understanding of program structures and fails to predict outputs or identify potential issues. Struggles to detect and resolve errors in programs, showing minimal progress in debugging. Demonstrates inadequate use of the programming language, with little adherence to programming practices, resulting in ineffective and poorly performing programs.

Student Rubrics

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

Assessment: Group Exercise

Criteria: Use of VBA
• **Excellent**: Demonstrates a thorough understanding and advanced use of VBA techniques. Effectively integrates relevant VBA functions, leveraging the language's features to optimize the solution. • **Good**: Utilizes VBA appropriately to address the task. Some elements may be implemented sub-optimally, but the code is functional and demonstrates a solid grasp of the language. • **Satisfactory**: Adequately uses VBA, but the solution may lack efficiency or sophistication. Several parts of the code may be overly simplistic or repetitive. • **Marginal**: Basic or incorrect use of VBA. The solution is incomplete or does not utilize appropriate functions and techniques. • **Fail**: No effective use of VBA, or the code fails to meet the required functionality.
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**: All requirements are fully met, with the solution exceeding expectations. The program is complete, addresses all tasks, and delivers robust functionality.
- **Good**: Most requirements are met, but there may be minor omissions or features that could be improved. The program performs the core tasks effectively.
- **Satisfactory**: Meets the basic requirements, but some key elements may be missing or inadequately implemented.
- **Marginal**: Significant requirements are not met, or the solution does not address major aspects of the task.
- **Fail**: Fails to meet the majority of the stated requirements.

Assessment: Assignment

Criteria: Use of VBA
• **Excellent:** (a) Demonstrates advanced proficiency in using VBA, employing appropriate loops (e.g., for-each loop vs. while loop) and error handling effectively. (b) Integrates VBA features such as named ranges and functions with clarity and precision. • **Good:** (a) Utilizes VBA well but may show some inconsistency in applying loops or error handling techniques. (b) Implements named ranges and functions, but the approach could be more refined. • **Satisfactory:** (a) Adequately applies VBA, though there may be minor errors or a lack of efficiency in using loops and error handling. (b) Uses basic features like named ranges but without optimal implementation. • **Marginal:** (a) Basic or incorrect use of VBA; loops or error handling are misapplied. (b) Limited or incorrect use of named ranges and functions, with noticeable issues. • **Fail:** Fails to demonstrate an understanding of VBA; loops, error handling, and named ranges are either missing or incorrectly used.
Criteria: User Friendliness
• **Excellent:** (a) Application is highly intuitive, with clearly labelled buttons and well-structured user interface. (b) Seamlessly integrates interactive features, ensuring ease of use for the target user. • **Good:** (a) Application is user-friendly with minor issues in button placement or interface structure. (b) Users can navigate the application with minimal difficulty. • **Satisfactory:** (a) Application is functional but may lack user-friendly features such as clear labels or easy navigation. (b) The interface requires some improvement to enhance user experience. • **Marginal:** (a) Application is somewhat difficult to use, with unclear buttons and confusing interface elements. (b) Requires significant improvement for better usability. • **Fail:** (a) The application is not user-friendly, with dysfunctional buttons or a poorly designed interface. (b) Difficult or impossible to navigate or use effectively.
Criteria: Meeting Stated Requirements
• **Excellent:** (a) Fully meets all stated requirements, including accurate output, correct calculations, and effective handling of dynamic data. (b) Demonstrates thorough understanding of business needs and addresses them appropriately. • **Good:** (a) Meets most requirements, with minor issues in output accuracy or handling dynamic data. (b) Some aspects may require further refinement or attention to detail. • **Satisfactory:** (a) Meets basic requirements but misses key elements such as accurate output or complete handling of dynamic data. (b) The solution works but is incomplete or lacks some necessary features.

• **Marginal:** (a) Partially meets the requirements with several aspects missing or incorrect. (b) The output is not entirely accurate, or dynamic data handling is insufficient. • **Fail:** (a) Fails to meet key requirements, resulting in incorrect output or failure to handle dynamic data. (b) Major parts of the task are not addressed or are entirely incorrect.
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:** Effectively applies advanced programming techniques to develop a well-structured, optimized solution. Demonstrates a deep understanding of the techniques and utilizes them appropriately throughout the program. The solution is both efficient and scalable. • **Good:** Applies advanced programming techniques with adequate structure and correctness. The solution meets the majority of requirements but may show slight inefficiencies or lack some optimization. • **Satisfactory:** Applies basic advanced programming techniques but may struggle with optimal use or show inefficiencies. The solution meets some requirements but lacks depth or coherence in technique implementation. • **Marginal:** Applies advanced programming techniques with limited effectiveness, resulting in suboptimal code. Some critical aspects of the solution are incomplete or poorly implemented. • **Fail:** Fails to apply advanced programming techniques effectively, resulting in an incomplete or incorrect program that does not meet the stated requirements.
Criteria: Effective Use of Programming Practices
• **Excellent**: Consistently demonstrates optimal programming practices, including clean, well-organized, and readable code. The program is maintainable, and conventions are adhered to effectively. • **Good**: Uses good programming practices with some minor issues related to structure, readability, or conventions. The program is functional but could benefit from improved organization. • **Satisfactory**: Basic adherence to programming practices, but the code may lack clarity or organization. Some areas could be improved for better readability or maintainability. • **Marginal**: Limited adherence to programming practices, with noticeable issues in code structure or readability. The program may be hard to follow or difficult to maintain. • **Fail**: Fails to apply standard programming practices, resulting in poorly structured, unclear, or unmaintainable code.
Criteria: Meeting Stated Requirements
• **Excellent**: Fully meets all stated requirements, demonstrating a comprehensive understanding of the task. The program is complete, accurate, and aligns perfectly with the specified objectives. • **Good**: Meets most of the stated requirements but may have minor deviations. The program is mostly accurate but could benefit from further refinement to meet all objectives. • **Satisfactory**: Meets some of the stated requirements but misses key elements. The program fulfils basic expectations but lacks some functionality or precision. • **Marginal**: Meets only a few of the stated requirements, and critical aspects are missing or incomplete. The program does not fully address the task. • **Fail**: Does not meet the stated requirements, with significant missing or incorrect functionality. The program fails to address the key objectives.

Assessment: Final Exam – Business Applications

Criteria: Solution Development and User Requirements
• **Excellent**: Develops a highly effective and innovative solution that thoroughly addresses the business challenge. Demonstrates a deep understanding of user needs and provides a well-structured, detailed approach that aligns with best programming practices. The solution is efficient and fully meets business and user requirements. • **Good**: Develops an effective solution that addresses the business challenge. Shows a solid understanding of user needs and provides a clear approach. The solution meets most of the requirements with minimal issues in efficiency or implementation. • **Satisfactory**: Develops a solution that addresses the business challenge but with some gaps or inefficiencies. The understanding of user needs is adequate, but the solution lacks detail or clarity in certain areas. The solution partially meets the requirements. • **Marginal**: Develops a solution that addresses the business challenge but is incomplete or significantly flawed. The understanding of user needs is superficial or vague. The solution partially meets the requirements but has major inefficiencies or errors. • **Fail**: Fails to develop a solution or provides an ineffective solution that does not address the business challenge. Shows limited or no understanding of user needs. The solution does not meet the stated requirements.
Criteria: Application of Programming Techniques
• **Excellent**: Applies advanced and fundamental programming principles to design an optimal business application solution. Demonstrates a strong grasp of program logic, structure, and efficiency, ensuring the solution is well-organized and performs efficiently. • **Good**: Applies relevant programming principles to design a solid business application solution. Demonstrates a good understanding of program logic, structure, and efficiency, with only minor issues in optimization or organization. • **Satisfactory**: Applies basic programming principles to design a business application solution. Some issues with program logic, structure, or efficiency are present, but the solution remains functional. • **Marginal**: Applies inadequate programming principles that result in a poorly structured or inefficient business application solution. Major issues with program logic or execution hinder functionality. • **Fail**: Fails to apply appropriate programming principles, resulting in a non-functional or ineffective business application solution. Significant issues with program logic and structure.
Criteria: Effective Use of the Programming Language
• **Excellent**: Demonstrates exceptional proficiency in using the programming language. Code is efficient, clean, and adheres to industry standards and best practices, ensuring optimal performance. • **Good**: Demonstrates good proficiency in using the programming language. Code is generally efficient and adheres to most best practices, with some minor deviations from optimal usage. • **Satisfactory**: Demonstrates basic proficiency in using the programming language. Code is functional but may lack efficiency or adherence to some best practices, resulting in less optimal performance. • **Marginal**: Demonstrates limited proficiency in using the programming language. Code is inefficient or poorly structured, with several deviations from best practices, affecting the performance and maintainability. • **Fail**: Fails to effectively use the programming language, resulting in inefficient, unorganized, or non-functional code that does not meet basic programming standards.

Criteria: Meeting Stated Requirements
• **Excellent**: Fully meets all stated requirements with a thorough and well-executed solution that is both complete and accurate. Demonstrates a strong understanding of the problem and its scope. • **Good**: Meets most of the stated requirements with a solution that is mostly complete and accurate. Some minor issues or omissions may be present, but the core requirements are addressed. • **Satisfactory**: Meets some of the stated requirements, but the solution is incomplete or contains significant inaccuracies or omissions. Addresses the main requirements but lacks depth or detail in certain areas. • **Marginal**: Meets few of the stated requirements, with major omissions or inaccuracies in the solution. The solution is incomplete and only partially addresses the core requirements. • **Fail**: Fails to meet the stated requirements or provides an incorrect or inadequate solution that does not address the core problem or requirements.

Efficient Email Communication Guidelines

To ensure prompt assistance, please include [**Course Code** - L**X**] (**X** being the section number), e.g., [**ISOM3230**-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 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

- Microsoft 365 (Windows) or Microsoft Office 2021 (for Windows)
- LLM, e.g., Copilot

Note: **Mac users are required to use Microsoft Excel (Windows version) during the class.**

Course Schedule

The course is offered in lecture sessions and laboratory sessions.

L1:	Monday	13:30 – 14:50	LSK 1005
	Friday	09:00-10:20	LSK 1005
LA1:	Friday	10:30 – 11:20	LSK G021
LA2:	Friday	18:00 – 18:50	LSK G021

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

Schedule of Lecture (Tentative)

Week	Date	Lecture	Assignment Due/Remark
1	2 Feb	Intro. to Course, Intro. to Programming	
	6 Feb	Intro. to Business Applications	
2	9 Feb	Macro Recording	
	13 Feb	VBA Basics	Add/Drop deadline: **Feb 14th**
3	16 Feb	VBA Basics	
	20 Feb	Input and Output	
4	23 Feb	Workbooks, Worksheets, and Error Handling	
	27 Feb	Ranges	
5	2 Mar	Ranges	
	6 Mar	If-then-else and select-case	
6	9 Mar	Looping	
	13 Mar	Looping	
7	16 Mar	Methods	
	20 Mar	Methods	
8	23 Mar	Practice	
	27 Mar	**Group Exercise**	
9	30 Mar	UserForm	**Assignment**: Release on **Mar 30th**
	3 Apr	**No Class**	
	6 Apr	**No Class**	
	10 Apr	Solver	
10	13 Apr	Arrays and Formulas	
	17 Apr	Arrays and Formulas	
11	20 Apr	NPV	
	24 Apr	NPV	
12	27 Apr	NPV	
	1 May	**No Class**	**Assignment**: Due on **May 2nd**
13	4 May	Business Application	
	8 May	Presentation and Revision	

(Note: Built-in Excel formulas are also covered in this course.)

Schedule of Laboratory (Tentative)

Lab No.	LA1(Fri)/LA2(Wed)/LA3(Thurs)	Topics
LA00	6 Feb	VBA Environment + Use of AI
LA01	13 Feb	Macro Recording + Excel Formulas and VBA Basics
LA02	20 Feb	Inputs and Outputs
LA03	27 Feb	Workbooks, Worksheets and Error Handling
LA04	6 Mar	Ranges + If-then-else and Select-case
LA05	13 Mar	Looping
LA06	20 Mar	Methods
LA07	27 Mar	TBA
LA08	3 Apr	**No class**
LA09	10 Apr	UserForm + Solver
LA10	17 Apr	Arrays and Formulas
LA11	24 Apr	NPV
LA12	1 May	**No class**
LA13	8 May	Revision

*** 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 Integrity

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.