

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. Lab Attendance	5%	As per course schedule
B. In-Class Exercise	10%	As per course schedule
C. Assignment	15%	As per course schedule
D. Midterm Exam	20%	To be announced
E. Final Exam — Advanced Techniques	25%	To be announced
F. Final Exam — Business Applications	25%	To be announced

A. Lab Attendance (5%)

Students will receive full marks for lab attendance by attending **any six** lab sessions, representing approximately 50%–60% of the scheduled lab sessions. Students who attend fewer than six lab sessions will receive a mark of zero for lab attendance.

B. In-Class Exercise (10%)

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

Each student is expected to apply their VBA programming skills independently to solve a business-related problem in Microsoft Excel (Windows version). At the end of the session, individual submissions will be collected and assessed. Students must submit their own work and demonstrate meaningful individual effort to receive credit.

Technical Requirements

Students are responsible for ensuring that Microsoft Excel (Windows version) is properly installed and fully functional throughout the in-class exercise period.

Late Submission Policy:

Late submissions for the in-class exercise within **10 minutes** will result in a **30% mark** deduction. Submissions **more than 10 minutes late will not be accepted**.

C. Assignment (15%)

The assignment requires students to apply VBA programming skills to analyze and solve business problems. Details of the assignment will be provided during the semester.

Late Submission Policy:

Late submissions for the individual assignment within **3 hours** will result in a **30% mark** deduction. Submissions **more than 3 hours late will not be accepted**.

D. Midterm Exam (20%)

The Midterm Exam assesses students' understanding of programming logic, syntax, and core VBA concepts covered up to that point. Please note that **no make-up opportunities** will be provided under any circumstances.

E. Final Exam — Advanced Techniques (25%)

This cumulative section of the final exam covers all lecture and laboratory materials taught during the semester. It assesses students' ability to apply advanced programming techniques across the course topics.

F. Final Exam — Business Applications (25%)

This cumulative section of the final exam covers all lecture and laboratory materials taught during the semester. It assesses students' ability to apply programming concepts in practical business contexts and develop solutions that address business needs.

Arrangements for the Make-up Final Exam

Make-up final examinations will be arranged only under exceptional circumstances beyond a student's control, such as a documented medical emergency. Students who are absent from the final examination due to a medical emergency must submit a completed and signed official make-up exam request form (**Form EX-16**), together with relevant medical documentation issued by a registered medical practitioner, to the course instructor by email before the University deadline. Such documentation is required before a make-up examination request can be considered. The make-up examination will be conducted in essay format.

(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.)

Remarks:

- 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.

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.

[In instances where a student is unable to check their paper within the stipulated checking period, the student's score will be deemed final by default. Regrettably, we won't be able to modify or rectify the score beyond the checking/appeal period.]

Mapping of Course ILOs to Assessment Tasks

Assessment Task	Mapped ILOs	Explanation
A. Lab Attendance	ILO1, ILO3, ILO4	Lab attendance supports students' regular engagement with programming practice, helping them apply programming techniques, develop and debug programs, and use the programming language effectively.
B. In-Class Exercise	ILO1, ILO2, ILO3, ILO4	The in-class exercises evaluate students' ability to apply programming concepts, analyze program logic, identify issues, and use VBA efficiently in a timed classroom setting.
C. Assignment	ILO1, ILO2, ILO3, ILO4	The assignment measures students' ability to design and develop a business application, analyze program structures, debug and refine their work, and follow sound programming practices.
D. Midterm Exam	ILO1, ILO2, ILO3, ILO4	The Midterm Exam assesses students' understanding of programming logic, core syntax, debugging concepts, and effective use of VBA in solving programming problems.
E. Final Exam — Advanced Techniques	ILO1, ILO2, ILO3, ILO4	This final exam component evaluates students' ability to apply advanced programming techniques, analyze complex program structures, debug errors, and optimize program performance.
F. Final Exam — Business Applications	ILO1, ILO2, ILO3, ILO4	This final exam component assesses students' ability to apply programming techniques in business application contexts, address user requirements, structure programs logically, and use VBA effectively.

Final Grade Descriptors:

Grades	Short Description	Elaboration on subject grading description
A	Excellent Performance	Demonstrates consistently excellent engagement across labs, in-class exercises, assignment work, and examinations. Applies fundamental and advanced VBA programming techniques with precision, analyzes program logic accurately, debugs effectively, and develops well-structured business application solutions that address user requirements.
B	Good Performance	Shows strong engagement and a solid understanding of programming concepts across the revised assessment tasks. Applies VBA effectively to business problems, demonstrates good program logic and debugging skills, and produces functional solutions with only minor limitations in efficiency, structure, or completeness.
C	Satisfactory Performance	Demonstrates adequate engagement and a basic understanding of VBA programming concepts. Can complete core lab, exercise, assignment, and examination requirements, but may show inconsistencies in program logic, debugging, advanced techniques, or business application design.

D	Marginal Pass	Demonstrates limited achievement of course outcomes. Engagement with labs and exercises may be uneven, and submitted or examined work shows only partial understanding of programming logic, debugging, advanced techniques, and business application requirements.
F	Fail	Demonstrates insufficient achievement of course outcomes across the revised assessment tasks. Shows inadequate engagement, limited understanding of VBA programming concepts, weak program logic and debugging ability, and an inability to develop effective business application solutions.

Student Rubrics

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

Assessment: In-class 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: Midterm Exam

Criteria: Programming Logic and Syntax
• Excellent: Demonstrates accurate understanding of programming logic, syntax, and core VBA concepts; answers are complete, precise, and well structured. • Good: Demonstrates solid understanding of programming logic and syntax with minor errors or omissions that do not significantly affect correctness. • Satisfactory: Demonstrates basic understanding of core concepts but shows some inconsistencies in applying programming logic or syntax. • Marginal: Demonstrates limited understanding of programming logic and syntax; responses contain significant errors or incomplete reasoning. • Fail: Fails to demonstrate sufficient understanding of core VBA concepts, programming logic, or syntax.
Criteria: Program Analysis and Debugging
• Excellent: Accurately analyzes program structures, predicts outputs, identifies errors, and explains debugging approaches clearly. • Good: Analyzes program structures and identifies most issues correctly, with only minor gaps in explanation or debugging detail. • Satisfactory: Provides basic analysis of program behavior but may miss some errors, outputs, or debugging steps. • Marginal: Shows weak analysis of program logic and limited ability to identify or explain errors. • Fail: Provides little or no meaningful program analysis and is unable to identify or explain major errors.

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 – 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 - LX**] (**X** being the section number), e.g., [**ISOM3230-L1**] 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.

Course Activity	Use of Gen AI Tools
In-Class Exercise	Allowed
Assignment	Allowed
Midterm and Final Exam	Not allowed
Lecture and Lab	Allowed
Outside the Class for Learning	Allowed

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)
- Access to approved generative AI tools, as permitted by the course policy

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 and Wednesday, 10:30 – 11:50, LSK1003

L2: Monday and Wednesday, 12:00 – 13:20, LSK1003

LA1: Tuesday, 09:00 – 09:50, LSKG021

LA2: Tuesday, 16:00 – 16:50, LSKG021

LA3: Friday, 09:00 – 09:50, LSKG021

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-Sep	Intro. to Course, Intro. to Programming	
2	7-Sep	Intro. to Business Applications	
	9-Sep	Macro Recording	
3	14-Sep	VBA Basics	
	16-Sep	VBA Basics	Add/drop deadline
4	21-Sep	Input and Output	
	23-Sep	In-class Exercise 1 Workbooks and Worksheets	
5	28-Sep	Ranges	
	30-Sep	Ranges	
6	5-Oct	If-then-else and Select-case	
	7-Oct	In-class Exercise 2 Looping	
7	12-Oct	Looping	
	14-Oct	Methods	
8	19-Oct	No class: Chung Yeung Festival	
	21-Oct	In-class Exercise 3 Practice	
9	26-Oct	Class Cancelled: Midterm Exam	Midterm Exam (Date TBC)
	28-Oct	Userform	
10	2-Nov	Solver	Assignment: Release on 2 Nov
	4-Nov	Arrays and Formulas	
11	9-Nov	In-class Exercise 4 Arrays and Formulas	
	11-Nov	Arrays and Formulas	
12	16-Nov	NPV	
	18-Nov	In-class Exercise 5 NPV	
13	23-Nov	NPV	
	25-Nov	Business Application	Assignment: Due on 24 Nov, 17:00
	30-Nov	Revision	

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

Schedule of Laboratory (Tentative)

Lab No.	LA1(Tue)/LA2(Tue)/LA3(Fri)	Topics
LA00	1, 4 Sep	VBA Environment + Macro Recording + Use of AI
LA01	8, 11 Sep	Excel Formulas
LA02	15, 18 Sep	VBA Basics
LA03	22, 25 Sep	Lab Question Session
LA04	29 Sep, 2 Oct	Workbooks and Worksheets
LA05	6, 9 Oct	Ranges
LA06	13, 16 Oct	Control Logic + Error Handling
LA07	20, 23 Oct	Looping
LA08	27, 30 Oct	Methods
LA09	3, 6 Nov	UserForm
LA10	10, 13 Nov	Arrays and Formulas
LA11	17, 20 Nov	Solver
LA12	24, 27 Nov	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.