



DESCRIPTION OF COURSEWORK

Course Code	CST 210
Course Name	Object-Oriented Programming Java
Lecturer	Toa Chean Khim
Academic Session	2023/09
Assessment Title	Assignment

A. Introduction/ Situation/ Background Information

This course provides a thorough introduction to the Java programming language. It will apply OOP principles to create a well-structured and maintainable codebase. It also leverage JavaFX to create an engaging and visually appealing game. It aims to impart the practical programming techniques to the students.

B. Course Learning Outcomes (CLO) covered

At the end of this assessment, students are able to:

CLO 4 Develop flexible, modular, and reusable event-driven GUI software.

C. University Policy on Academic Misconduct

1. Academic misconduct is a serious offense in Xiamen University Malaysia. It can be defined as any of the following:
 - i. **Plagiarism** is submitting or presenting someone else's work, words, ideas, data or information as your own intentionally or unintentionally. This includes incorporating published and unpublished material, whether in manuscript, printed or electronic form into your work without acknowledging the source (the person and the work).
 - ii. **Collusion** is two or more people collaborating on a piece of work (in part or whole) which is intended to be wholly individual and passed it off as own individual work.
 - iii. **Cheating** is an act of dishonesty or fraud in order to gain an unfair advantage in an assessment. This includes using or attempting to use, or assisting another to use materials

that are prohibited or inappropriate, commissioning work from a third party, falsifying data, or breaching any examination rules.

2. All assessments submitted must be the student's own work, without any materials generated by AI tools, including direct copying and pasting of text or paraphrasing. Any form of academic misconduct, including using prohibited materials or inappropriate assistance, is a serious offense and will result in a zero mark for the entire assessment or part of it. If there is more than one guilty party, such as in case of collusion, all parties involved will receive the same penalty.

D. Instruction to Students

This assignment is a **Group assignment (1/2 persons per group)**. Each student should submit a zip file "**Group_StudentName_Assig.zip**" to Moodle. The submission should include the following:

1. The **PDF** should include the **Assignment Cover Page, Screenshots of source code with comments, and Marking Rubrics**. Renamed as **< StudentName _PDF>**. (Notes: Please include your members name in Cover Page)
2. The **project folder (Source code)**. Export your source code into **Compressed (zipped) folder**.
 - If you are using the Eclipse, go to File → Export → General → Archive File → Next → Tick the folder → Click Browse → find your specific path and named to **< StudentName _code>** → Click Finish. Your file should be being zipped in your specific folder.
3. Take a **short video** of your GUI outcome and save as **<StudentName_Demo>.mp4**.
4. The submission deadline of Assignment 1 is on **23:55, 18 December 2023**. Overdue penalty will be given to the assignment that is submitted after the deadline (20 marks deducted for each day).
5. Submission type: Softcopy

***Note: Only JavaFX is acceptable.**

E. Evaluation Breakdown

No.	Component Title	Percentage (%)
1.	Task(s)	100
	TOTAL	100

F. Task(s)

In this assignment, your task is to design and implement a graphical game of your choice using Java and JavaFX, while adhering to Object-Oriented Programming (OOP) principles. The game you create can be of any genre or type, allowing you to express your creativity and explore various aspects of OOP in game development.

Requirements:

1. Game Design:

- Choose a game concept that interests you. It could be a classic game, an original creation, or a variation of an existing game.
- Create a design document outlining the game's rules, objectives, as well as the game's graphical elements.

2. Class Design:

- Implement a class structure for your game, following OOP principles. This should include classes that represent game elements, such as characters, objects, levels, and graphical components.
- Utilize inheritance, polymorphism, and encapsulation where appropriate.
- Include attributes and methods that are relevant to your game's functionality.

3. JavaFX Graphics:

- Use JavaFX to create the graphical user interface for your game.
- Incorporate graphics, animations, and interaction elements to enhance the player's experience.

4. Game Logic:

- Implement the core game logic, including player interactions, scoring, game state management, or win/lose conditions.
- Ensure that the game provides meaningful graphical feedback to players.

5. User Interface and Controls:

- Design and implement user-friendly controls and a graphical user interface that allows players to interact with the game.
- Include buttons, menus, and user prompts as needed for a seamless gaming experience.

6. Documentation

- Include comments and documentation within your code to explain the purpose and usage of each class and method.
- Provide instructions for playing your game.

As part of this assignment, you will be required to demonstrate your GUI program. The demonstration should cover the features of your program, including its design, functionality, and creativity. The presentation should be delivered in front of the class after done assignment submission.

APPENDIX 1

MARKING RUBRICS

Component Title	Task(s)					Percentage (%)	100
Criteria	Score and Descriptors					Weight (%)	Marks
	Excellent (5)	Good (4)	Average (3)	Need Improvement (2)	Poor (1)		
OOP Implementation	OOP principles are expertly applied, with a well-structured and maintainable codebase. Inheritance, polymorphism, and encapsulation are used effectively.	OOP principles are correctly applied, with a mostly well-structured and maintainable codebase. Inheritance, polymorphism, and encapsulation are used appropriately.	OOP principles are applied to some extent, with a partially well-structured codebase. Inheritance, polymorphism, and encapsulation are used but with issues.	OOP principles are minimally applied, with a disorganized and hard-to-maintain codebase. Inheritance, polymorphism, and encapsulation are lacking.	OOP principles are not applied, leading to an unstructured and unmaintainable codebase. No use of inheritance, polymorphism, or encapsulation.	20	
Functionality & Completeness	The game functions perfectly as intended, with all core mechanics implemented and no issues.	The game functions well, with minor issues or improvements needed in some core mechanics.	The game functions satisfactorily but has notable issues or missing core mechanics.	The game has significant functionality issues, with major missing core mechanics.	The game is non-functional or barely functional, with no core mechanics implemented.	20	
Code Organization & Readability	The code is exceptionally well-organized, easy to follow, and consistently styled.	The code is well-organized, generally easy to follow, and mostly consistently styled.	The code is somewhat organized, moderately easy to follow, and inconsistently styled.	The code is disorganized, hard to follow, and inconsistently styled.	The code is extremely disorganized, virtually impossible to follow, and unstyled.	20	
Comments and Documentation	Thorough comments and documentation, explaining the purpose and usage of each class and method in detail. Clear instructions for playing the game.	Good comments and documentation, providing adequate explanations and instructions.	Basic comments and documentation, lacking some details and instructions.	Minimal comments and documentation, missing essential explanations and instructions.	No or negligible comments and documentation, leaving the code and game unexplained.	20	
Concept	The game concept is innovative and engaging, demonstrating creativity and originality.	The game concept is interesting and creative, showcasing an engaging idea.	The game concept is somewhat engaging, but lacks significant innovation or creativity.	The game concept is average and not particularly engaging or creative.	The game concept lacks innovation, creativity, and engagement.	10	
Individual Contribution	Exceptional individual contributions, demonstrating high-quality work and a strong sense of responsibility.	Good individual contributions, demonstrating solid effort and contributions from all team members.	Adequate individual contributions, demonstrating satisfactory effort and contributions from most team members.	Limited individual contributions, indicating minimal effort or inconsistent contributions.	Very poor individual contributions, indicating no or negligible effort and minimal or no contributions.	10	
TOTAL						100	

Note to students: Please include the marking rubric when submitting your coursework.