FACULTY OF ENGINEERING AND TECHNOLOGY

FINAL ASSESSMENT FOR THE BSC (HONS) INFORMATION TECHNOLOGY; BSC (HONS) COMPUTER SCIENCE; BACHELOR of SOFTWARE ENGINEERING (HONS)YEAR 2

ACADEMIC SESSION 2025; SEMESTER 3

PRG2104: OBJECT ORIENTED PROGRAMMING

Project **DEADLINE: Week 14**

INSTRUCTIONS TO CANDIDATES

- This assignment will contribute 50% to your final grade.
- This is an individual assignment.

IMPORTANT

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

- Coursework submitted after the deadline will be awarded 0 marks
-

Lecturer's Remark (Use additional sheet if required)

I..... (Name) .....std. ID received the assignment and read the comments..... (Signature/date)

Academic Honesty Acknowledgement

"I .....(student name). verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties *(refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme)* for any kind of copying or collaboration on any assignment."

…..... (Student's signature / Date)

Overview

The aim of this project is to achieve the learning outcomes (CLO 4) of this subject as mentioned in the syllabus, your role is to analyse, apply, and design a software application using object oriented programming. You also need to demonstrate your work at the time of submission. This overall assignment mark will contribute 50 % of your final grade.

ASSIGNMENT SPECIFICATION

Learning Outcome Being Assessed	Write computer programs that utilise third party object oriented libraries. (C3, PLO 7)
Submission Deadline **Microsoft Team Submission**	**Monday, (Week 14) by 4.00p.m.** Late submission will be awarded 0 mark. Create a submission folder named "Project_yourID". Put your project folder and documentation report into this submission folder. Zip it and submit this zipped file to the Microsoft Team. In Microsoft Team, the zip file can be attached to the assignment page with a softcopy of reports
Git Hub Assignment Link	https://classroom.github.com/a/ZQHvvCOj **Please use the assignment link for code versioning. The record in git repository will be used to support originality of the work.**
Outline of Problem	This assignment stipulates the design of a system by identifying the required classes and their relationships. Students are required to demonstrate the ability to apply their knowledge of inheritance and polymorphism in their implementation of the solution. Students also need to make use of the scalaFX GUI library to create a GUI Application. Other GUI libraries are not acceptable.
Detail Question	Propose a standalone GUI system that helps achieve United Nations Goal 2: End hunger, achieve food security and improved nutrition, and promote sustainable agriculture. This system should make use of object-oriented programming features. Students should propose additional classes or traits to achieve a good design that reduces code repetition. Students are required to design their user interface for ease of use.

	The complexity of the system will determine the grade of the project. The more you demonstrate your object-oriented skills, the higher your possible grade will be, which aligns with the uniqueness and originality of your program. The following are examples of projects you can explore: • **Food Distribution Management**: Track and manage the distribution of food supplies. • **Nutritional Information Database**: Provide information on the nutritional content of various foods. • **Agricultural Resource Management**: Manage resources for sustainable agriculture. • **Community Engagement Platform**: Facilitate communication and collaboration among community members. Attention: If you refer to any sources from youtube, github or other code repositories in creating the application, please do cite. Any works that are not created by you will not be considered in marking. If you refer to sources from your senior, you will be caught for plagiarism. The lecturer will check and compare your code to the previous submission library.
Development Platforms	The technologies that you should be using are: 1. SBT builder (Any Latest Version) 2. Any IDE of your choice 3. Java 21 Library 4. Scala 3 Library 5. ScalaFX 21 (other GUI libraries are not allow)
What you should hand in	The following items are to be handed in your report in softcopy: • A cover page (use the template provided). • A documentation report includes the UML diagrams that describe the classes/objects identified from the problem domain, and their relationships among these classes/objects used in the program. – in softcopy • A description that shows the program is working for all 4 features. • Your GitHub Account username, which you used to enrol in the GitHub assignment • An A4 page to be written by **you on the** personal reflection that includes: o A description of how you applied the above object-oriented concepts in your assignment.

	○ The problems encountered during this assignment and how you solved these problems. ○ An evaluation of the strengths and weaknesses of your submitted work. • The project/solution files, including the source code and application program. – in **softcopy** • A video that shows the running of your application. NOTE: Submitting the assessment means you have agreed that your work is original and comply with the rules and regulations (refer to Academic Impropriety)
Paper Size / Format	Paper size A4 (Use only one side of the paper) For the personal reflection write-up, Paragraph format 1.5-line spacing Font size 12 points
Academic Impropriety	Sunway University takes a strong stand on plagiarism. Any students found to have copied work, colluded or presented work that is not their own will be punished under the terms stated in the rules and regulations booklet. Students are permitted to use 3rd party components, however all such code must be well described and credit awarded to the respective owner. Students must also ensure that the majority of source code is their own, and that the core algorithms are their own work. The use of copyright materials is forbidden. *subject to change anytime without prior notification **The work that you submit must conform to those regulations.**
Assessment: **Report**	Contributes 50% to the overall final assessment mark. Refer to ASSESSMENT CRITERIA FOR Project table for further elaboration of marking distribution.

ASSESSMENT CRITERIA FOR Project

Mark / General Impression	Area / Assessment Criteria							
	Class Definition and Design			**Application Program**		**Style**	**Use of Third-Party Library**	**Documentation**
	Fulfillment of Requirements (x2)	**Relationships among classes**	**UML Class Diagram**	**Fulfilment of requirements (x2)**	**GUI Implementation**	**Naming Convention**	**Mastery**	**Documentation (Report)**
5 Excellent	(a) Correct and complete • Classes – both basic classes and the "collection" class • Identification of data fields, visibility modifiers and types. • Constructors (b) Additional features / operations provided. (c) Originality and Uniqueness. (d) Github record provide very good confidence of originality.	Correct application of all the following concepts: • inheritance • polymorphism • abstract class • generic programming.	Perfectly correct diagram. • All notations are correctly used. • Class members are complete, and • The diagram is consistent with the class design.	The following are provided • Menu navigation • Execution of all the required operations • Input validations • Completely correct, efficient and elegant use of programming constructs. • Methods are extensively used to achieve complete modular programming.	Illustrated an excellent mastery in Event Driven Programming and correct use of layout classes. GUI Components and layout component are design and use correctly.	Full adherence to naming convention with appropriate, meaningful and correctly spelt identifier names.	An excellent correct use of third-party libraries to solve problem. No errors in utilizing the third-party libraries. Demonstrated in depth understanding of third-party library model.	• Complete and well written documentation. • All Required section is included. • Table of content is formatted properly. Very few typo or spelling mistake.
4 Very Good	Correct and complete • Classes – both basic classes and the "collection" class • Identification of data fields, visibility modifiers and types. • Constructors • Quite original and uniqueness	Correct application of the following concepts: • inheritance • polymorphism • abstract class	• Complete with only one very minor error in notations used. • Diagram is consistent with class design.	The following are provided • Menu navigation • Execution of all the required operations • Correct and efficient use of programming constructs.	Illustrated a good mastery in Event Driven Programming and correct use of layout classes. GUI Components and layout component are design and use	Adherence to naming convention with meaningful, appropriate and correctly spelt identifier names.	Make a good use of some third-party libraries to solve problem. No errors in utilizing the third-party libraries.	• Complete and well written with only one very minor error in documentation. • Not all required section is included only miss one section.

	• Github records gives good confidence of originality.			• Methods are used to achieve a high degree of modular programming.	correctly with minor error.		Demonstrated in good understanding of third-party library model.	• Table of content is formatted reasonably. Few typo or spelling mistake.
3 Average	Correct and complete • Basic classes • Identification of data fields, visibility modifiers and types. • Constructors • Partial originality and uniqueness • Github records give moderate confidence of originality.	Correct application of the following concepts: • inheritance • polymorphism	Quite complete with not more than two minor errors in notations used. Diagram is consistent with class design.	The application program demonstrates the correct execution of all the required operations One or two minor errors.	Illustrated an average mastery in Event Driven Programming and correct use of layout classes. GUI Components and layout component are design and use correctly with some error.	General adherence to the naming convention with one or two minor errors.	Make use of some third-party libraries to solve problem. Minor errors in utilizing the third-party libraries. Demonstrated in average understanding of third-party library model.	• Quite complete and good written with only few minor error in documentation. • Not all required section is included only miss few section. • Table of content is formatted. Few typo or spelling mistake.
2 Poor	Some errors or 1 incomplete basic class. No originality with some uniqueness Github records give poor confidence evidence of originality.	Correct application of inheritance	Some errors in notation or diagram's consistency with class design.	Incomplete application program. Some errors.	Illustrated a poor mastery in Event Driven Programming and correct use of layout classes. GUI Components and layout component are design and use correctly with few error.	Limited adherence to naming convention.	Make use of a third-party libraries to solve problem. Some errors in utilizing the third-party libraries. Demonstrated in poor understanding of third-party library model.	• Incomplete in documentation with few major error. • Not all required section is included with missing lot of section. • Table of content is partially formatted. Lot of typo or spelling mistake.
1 Very Poor	Very major errors or more than 1 incomplete basic class.	Illogical inheritance hierarchy	Major errors in notation or diagram's	Grossly incomplete application program.	Illustrated a very poor mastery in Event Driven Programming and	Serious lack of adherence to	Make use of a third-party libraries.	• Grossly incomplete in documentation with major error.

	Github records give very poor confidence of originality.		consistency with class design.	Very major errors.	incorrect use of GUI components.	the naming convention.	Major errors in utilizing the third-party libraries.	• Major required section is not included. • Table of content is not formatted. Lot of typo or spelling mistake.