UTS College

Project A

Diploma of IT
Programming 2

SEMESTER 2 2023

Weight: 20%

DUE : Week 10

ASSESSMENT TASK 1

TASK DESCRIPTION

This project requires students to create an object oriented design and coded solution for a chosen application, using all the concepts taught in the subject up to and including week 9.

You may choose to complete this assessment with at most one (1) other person. To work as a pair; you and your partner MUST be in the same tutorial class. You will need to spend approx. 10 hours together working on the assessment (outside of tutorial hours). If you do not intend to spend this time working together – choose to complete the assessment on your own. Both partners must put in an equal amount of work and effort, or the group will be dissolved by the tutor.

OBJECTIVES

1. Evaluate if a software solution is well-designed and fit for purpose.
2. Design a well-constructed object oriented solution from a specification.
3. Demonstrate a working knowledge of list and map data structures.
4. Use inheritance to improve the system design.
5. Construct a GUI interface

UTS College

BRIEF

This is a take home assessment to evaluate your understanding of Java programming concepts taught so far in IPRO002. The assessment will be composed of three (3) parts: the scenario, the code and explanation (including proof of testing).

THE SCENARIO

You will use your Project A as the starting point for this project.

You will design a Graphical User Interface to support your application, using the domain classes from Project A.

Your GUI design should outline the main use cases for the application, the basic pane design and situations to handle for the application. Use a main window and sub windows/forms.

You will design the windows and panes for your application. You will implement all the GUI components following your design. You will implement the MVC design pattern and test that your solution satisfies the requirements of your application.

GUI DESIGN

Create your design for the GUI components – think about input needed from the user. Make sure your GUI design shows the GUI layout for your application. You can use paint or word tools to create the design. Make sure each panel/pane has at least 1 button to handle events.

THE CODE

Copy your project A into a new directory. Add 3 Java files corresponding to the Model, View and Controller components of MVC. Delete the In class and change all dependencies to use parameters for input.

Write your GUI components. Make sure you are using the MVC pattern and data binding and/or property listeners to refresh each window when the data changes in the Model.

Be sure to adhere to standard Java naming conventions.

As with Programming 1, you must only use code constructs taught in weeks 1 to 9 inclusive. Using constructs outside of week 1 to 9 will lower the marks you receive.

THE EXPLANATION (300 WORDS)

Write a detailed explanation of how the user would interact with your GUI to perform each of the use case tasks. Explain how the MVC pattern works by following the flow of control for one of the use cases. You should place comments **in the code** to identify the steps in the MVC pattern. Include any issues you had and how you resolved them. Minimum 300 words.

UTS College

SUBMISSION

You will submit your assessment in MS Word document format – yourName-studNo.doc.

The document will have 3 sections:

GUI Design: Your designed GUI including buttons, panels, labels and textfields to support your application.

The code: You will copy the code from VS Code into this section

The explanation.

You will also submit a copy of your VS Code project as a zip file

ASSESSMENT CRITERIA

CRITERIA	WEIGHT	SLOs / LLOs	PLOs
Functional specifications	**35%**	**2, 3**	**A1, B1, E1**
Code layout	**35%**	**1, 3, 4, 5**	**A1, B1, E1**
Explanation	**30%**	**4, 5, W4**	**C1**

SLOs: subject learning outcomes
PLOs: program learning outcomes
LLOs: language learning outcomes

Please refer to the next page for the rubric definition for each criteria and grade.

UTS College

		HD	D	C	P	F3	F2	F1	F0
1	Functional specifications	The program works correctly in every way and meets all of the functional specifications using MVC architecture. The interface is very intuitive and easy to use.	The program runs and produces mostly correct results using MVC architecture. It also meets most of the other specifications. Some error checking may be missing. The interface is intuitive and easy to use.	The program produces mostly correct results using MVC architecture. Some minor functionality may be missing. The interface is somewhat intuitive.	The program is producing some incorrect results or a part of the program is incomplete. Some MVC architecture is present but may not be working correctly. The interface isn't intuitive to use.	The program is producing mostly incorrect results or a most of the program is incomplete.	The program is producing all incorrect results or mostly incomplete.	The program does not run; Some code is valid	Program does no required tasks
2	Code layout	The code is exceptionally well organised, flow of control is very easy to follow and variable names chosen clearly explain what the code is accomplishing and how. No design rules or	The code is easy to follow and well organised and variable names chosen are very useful in understanding what the code does and how it is accomplished. Some minor design rules or	The code is fairly easy to follow and variable names chosen are useful in understanding the code. Some design rules or patterns have been broken. Java naming	The code is a little hard to read and variable names chosen are somewhat useful in understanding the code. The flow of control is difficult to follow. Java	The code is hard to read and variable names chosen at least once help understand what they are used for. Java naming conventions not used.	The code is poorly organised and very difficult to follow. Variable names chosen do not help understand what they are used for. Reader must guess what variables are for. Java naming conventions not used. Only 1 Java construct is from week 1 to 9.	The code is poorly organised and very difficult to follow. Variable names chosen use names that will not compile e.g. with spaces. Java naming conventions not used.	No valid code written

UTS College

		patterns have been broken. Java naming conventions always used. All Java constructs are from week 1 to 9.	patterns have been broken. Java naming conventions mostly used. Most Java constructs are from week 1 to 9.	conventions sometimes used. Java constructs are generally from week 1 to 9.	naming conventions used at least once. Some Java constructs are from week 1 to 9.	A few Java constructs are from week 1 to 9.		No Java construct is from week 1 to 9.	
3	Explanation	Demonstrates a clear and comprehensive understanding of GUI concepts. The code contains comprehensive comments that explain steps in the MVC pattern	Demonstrates a clear understanding of GUI concepts. The code contains many comments that explain steps in the MVC pattern	Demonstrates a moderate understanding of GUI concepts. The code contains some comments that explain steps in the MVC pattern	Demonstrates little understanding of GUI concepts. The code contains a few comments that explain steps in the MVC pattern	Demonstrates very little understanding of GUI concepts. The code contains a couple of comments that explain steps in the MVC pattern	Demonstrates almost no understanding of GUI concepts. The code contains no comments that explain steps in the MVC pattern	Demonstrates almost no understanding of GUI concepts. The code contains no comments that explain steps in the MVC pattern.	No Sub-mission