Comp 1502 F15 **Assignment 2 Marks**

Name: ______________________________

Assignment Solution	***75 / 75***	***Performance Requirements***
❐ **User Interface Class**	**5 / 5**	**program properly uses the provided UserInterface class** ▪ Used for all input to the program ▪ If required methods have been added to extend the provided services of the class o Methods match the style and purpose of the class.
❐ **Proper use of main**	**5 / 5**	**proper use of instantiated objects** ▪ Program is launched with a static main method in an appropriately named class ▪ All significant processing is performed in instantiated objects
❐ **Loading and holding data**	**12 / 12**	▪ File name is retrieved from the first command line argument • Proper error checking to confirm the file name • A proper error message and proper program termination o **System.exit() is not allowed** ▪ File has been properly opened • Basic error checking if the file does not exist on the system (as shown in class examples) ▪ Input of question information from the file • Number of questions read in • Array set up to store information using the number of questions read in • Loop to read in all questions o Proper information for a question loaded and stored into a Question object o Object has been stored into an array • All information has been loaded from the file into an array of objects before the game starts ▪ Basic file error checking • Handles case of expected line is missing • Handles case of the Points, answer, number-of-answers missing
❐ **Trivia Game Play**	**36 / 36**	**Some marks in this section include design of how you accomplished the task**
	1 / 1	❐ Program starts up in the main
	3 / 3	❐ Questions have been presented in the proper order (as retrieved from the file) for the entire game.
	1 / 1	❐ Questions presented one at a time
	4 / 4	❐ Proper header printed above each question ▪ The current question number ▪ The total number of questions ▪ The possible points for the question ▪ The user's current score.
	3 / 3	❐ Question presented to the user with all possible choices **(including skip and quit)**
	3 / 3	❐ User can enter choice for their answer or to skip or quit (error checking also done for wrong values)
	6 / 6	❐ Proper verification of answer from the user ▪ If they got it wrong or right (message displayed) ▪ All stats properly updated ▪ Steps to the next question
	5 / 5	❐ Skip question works properly in all cases (including final stats)
	5 / 5	❐ Quit game works properly in all cases (including final stats)
	5 / 5	❐ Ends the game automatically once they have finished the last question (even if they skip the last question)
❐ **Trivia Game Flow**	**10 / 10**	**Feedback on the quality of the game play code. Points considered:** ▪ Division of tasks, The flow of method calls in the program (Do they make sense), are they clearly defined tasks that are, Is it a clean design? ▪ This value gives a ranking of your overall design. (You can convert it to a percentage and look to the chart on the first page to see where your program flow lands)
Comment:		

❒ **Final Game Statistics** 4 / 4

- ❒ Clean nice output with labels making it clear what information the user is seeing
- ❒ All stats have been properly calculated
- ❒ The final score /points the user received (would be nice to have the total number of points possible as well)
- ❒ The total number of questions answered correctly is displayed
- ❒ The total number of questions skipped is displayed
- ❒ Presented before the program exits (after the finish of a game or the user has selected to quit)

❒ **User Interaction** 3 / 3

- ❒ Output
  - Meaningful messages and output given to the user
  - Nicely formatted output with clear labels for the user to understand
  - Proper error messages for incorrect actions the user does or failures in the program.

Code Design ***25 / 25*** ***Quality and Readability Requirements***

❒ **Readability** 5 / 5

- overall, code is clear and concise
- all identifier names are self-documenting and follow course naming conventions
- hard-coded magic number literals are avoided in favour of named constants best declared at the top of a class in all capitals
- all code is correctly and consistently indented
- all code is formatted to enhance readability; white space is inserted around logically-related code blocks
- tricky or less obvious sections of code are accompanied by short clarifying comments, as appropriate (this will be weighted heavier for marks)

❒ **Design** 20 / 20 **Design and other concerns**

10 / 10 ❒ clearly-identifiable subtasks are delegated to clearly-named helper methods
- Avoidance of large over complicated methods
- Avoidance of too simple of methods where they are not really needed (one line methods that are only called once)
- code duplication is avoided; algorithms needed more than once are placed in helper methods that are called as needed

3 / 3 ❒ proper use of methods in the class (3 = good, 2 = needs work, 1 = needs a lot of work)
- includes parameter passing and returning information from a method (only use instance varaibles when they are needed in >1 method)

2 / 2 ❒ proper choice of which methods are in each class (2 = good, 1 = needs work, 0 = needs a lot of work)
- Should make sense based upon what the class is supposed to represent (its responsibilities)

1 / 1 ❒ all methods are marked public or private, as appropriate

1 / 1 ❒ all instance variables are marked private (missing even one will result in zero for this mark)

3 / 3 ❒ Choice of instance variables versus local variables is appropriate. Does it make sense for the lifetime of the class it is declared in. Could it have been a local variable in each method it is used in. Could it have been a parameter or return from method instead. (3 = good, 2 = needs work, 1 = needs a lot of work)

Subtotal **100 / 100**

Deductions

Use of static key word other than for the main method (up to -20 marks)
-1 for every instance variable not marked private (Max of 5 marks taken off)
-1 for very helper method that is not marked private (Max of 5 marks off)
- 30% if program uses ArrayList instead of Array
-3 program does not format all decimal values outputted to 1 decimal places
Other:

TOTAL **100 / 100**

Mark 5.0 / 5

Comments: