CM1601 Programming Fundamentals Assignment Specification (2020)	
Module Leader	Rajitha Jayasinghe
Unit (Group/Individual)	30
Weighing	20%
Qualifying Mark	40% (both parts)
Learning Outcomes Covered in this Assignment:	LO2. Design, code, compile, test and execute fundamental programming concepts using a high-level programming language. LO3. Construct robust, maintainable programs that use object-oriented analysis and design principles
Handed Out Date	31st May 2021
Due Date	12th July 2021
Expected Deliverable	Source code and report
Method of Submission	Online
Method of Feedback and Due Date	21st July 2021
BCS Criteria Met by this Assignment	

Assessment Regulations

Refer to the "How you are assessed section" in the Student Handbook for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

Coursework received late without valid reason shall not be accepted and shall receive no grade, but shall count as one of the assessment opportunities prescribed in paragraph 9 of **RGU Academic Regulation A4 section 4.3.**

It is recognized that on occasion, illness, personal crisis or other valid circumstances can mean that you fail to submit and/or attend an assessment on time. In such cases you must inform the School of any extenuating circumstances through a ***Coursework Extension Form*** or a ***Deferral Request Form***, with valid evidence for non-submission of an assessment up to a maximum of five working days after the assessment submission date. This information will be reported to the relevant Assessment Board that will decide whether a student should be allowed to reattempt without penalty (a deferral). For more detailed information regarding University Assessment Regulations and accessing forms, please refer to the following website: www.rgu.ac.uk/academicregulations www.rgu.ac.uk/academicregulations

Grading

Marks will be awarded for the coursework based on the provided Grading Grid. These marks will be mapped onto a grade scale from A-F as determined by the individual module coordinator

Note:

Subject to approval by external examiner

Coursework Specification

Objective

Please note that this is a continuation of Coursework1. You are instructed to use Java and OOP concepts to develop the Tower of Hanoi game which you have developed earlier.

But this time the specification has changed slightly.

- Program needs to have a user interfaces and entering information, displaying moves and all the other functionalities needs to operate via an interface.
- Unlike the previous application this application needs to record player's information such as name, difficulty level
- A special windows to display the highest scorers. This is similar to a scoreboard of a game
- After the player finished the game successfully and if the player completes the whole game without the help of fast forward option, final number of moves and the player's information need to be saved in a file.
- Other than the above factors, the original specification remains unchanged.

Tasks

1. All the functionalities which were developed earlier in your last coursework need to be implemented again. You are allowed to use Java FX for implementing the user interfaces. You have the freedom to design the interfaces. When implementing the application, you are instructed to use classes and objects with OOP concepts for
    a. Classes and objects: Mandatory to use
    b. Encapsulation : Mandatory to use
    c. Inheritance: Optional
    d. Abstraction and polymorphism: Optional
2. You are free to choose the data you want to save in the file. Do not use a database to store information
3. Create a Junit test suit to cover the whole scenario. All test cases need to have a proper description, expected, actual outputs and the status of the test case. Also attach the test plan to your report (LO2 and LO3).
4. Explain what you have done in order to ensure the robustness and the maintainability of your source code. Mainly explain whether you followed the guidelines you were taught during the code quality lecture sessions. Further you have followed SOLID principles when designing the system or not. Feel free to refer online resources when justifying (LO3).

Marking schemes

Description	Marks
Implementation of the algorithm	

(if the student uses at least mandatory OOP requirements, full marks for each requirement =5, if not half marks=2.5) • Tower of Hanoi algorithm • Deals with the user information + Scoreboard • Fast forward + previous step feature • Feedback in expert level	 5 5 5 5
User interface	5
File handling	3
Junit and Testplan	5
Justification of the maintainability and robustness. - If the justification is reasonable and it was followed when implementing the source code at least for some extent - If the justification is reasonable but not applied during the implementation	 4 or 2
Additional improvements (GUI creativity, OOP inheritance, abstraction and polymorphism usage. separation of the user interface and the algorithm properly)	5
Viva	8