Rubric for Marking Software Engineering Fundamentals Assignment 2019 Semester 1

Distribution of Marks: Assignment Marking

1. **Progress Marks (10)**
2. **2 Milestones (2 x 5)**
3. **Face-to-face marking (20 marks)**

1. Progress Marks (10 marks)

Wk	Suggested Activities (these may vary between teams but students are expected to show progress)	Progress	Milestone
3	Select the project and form the team – consider the strengths, interests and experiences of team members	1 mark	
4	Identify User stories for Product/ Release Backlog	1 mark	
5	Identify main domain classes and add responsibilities such that objects can interact to get the first release done. Distribute classes/responsibilities to members.	1 mark	
6	Write Test Cases and start writing code to make them pass (individual)	1 mark	
7	**Initial Milestone**: Present partially completed product and get feedback from tutor and class. Demonstrate working Test cases (individual) and continue integrating code and testing.	1 mark	5 marks (see section 2)
8	Plan second release incorporating feedback. Add classes, extend class responsibilities and distribute work.	1 mark	
9	Develop detailed Group UML Diagrams (class, use-case).	1 mark	
10	Develop Individual UML Diagrams (sequence, state, object) Demonstrate individual Test cases	1 mark	
11	Develop Activity Diagrams. Combine code and run integrations tests. Resolve conflicts. Refactor design and code before final submission.	1 mark	
12	**Second Milestone**: Present final product to tutor and class.	1 mark	5 marks (see section 2)
13	**Face to Face Marking of design and code by Class Tutor**		20 marks (see section 3)

2. Milestones 1 and 2

Milestone 1 (Total 5 marks) in Week 7

This first demo/presentation will award marks for both group and individual work. Each individual member is expected to undertake at least two major tasks or responsibilities. Some of these responsibilities can be shared. For example, in the chess-like game “show all valid move destinations for a piece” can be a user-story/responsibility that must be shared by the Piece and the Board classes. The subclasses of Piece such as Rook and Knight may know valid moves but only the Board classes knows where the vacant and opponent squares are located. In such a situation, those members writing the Rook and Knight classes may override the method getAllValidMoves():Squares[], while the person developing the Board class may write a method to check which of these squares are free to move using IsValidDestination(square). These two methods from different classes may thus be combined to determine all valid destinations. Students can defer non-model (GUI) aspects until the “core-logic” is fully tested (by using a simple textual interface).

Activity	**Marks**
Group Part	
Presenting the Initial Release Backlog (design documents: Class and Use Case diagrams) and demonstrating a partially completed product	2 marks (4 mins)
Individual Part	
1. Individual member is expected explain his or her role/obligation towards carrying out the main user-stories/responsibilities, and any challenges faced. 2. Individual member is expected to demonstrate how these obligations are discharged, by running some test cases. If code implementing these tests is not complete, present the test cases designed. If little or no progress has been made briefly explain the reasons.	3 marks (4 x 1.5 mins)

Milestone 2 (Total 5 marks) in Week 12

The second demo/presentation will award marks for both group and individual work though more marks is awarded for group work.

Activity	**Marks**
Group Part	
Presenting the Final Release Backlog and demonstrating the final product	2 marks (3 mins)
Use a class diagram to present the overall design; use two sequence-diagrams to present the core functionality. Explain the design rationale highlighting usability, maintainability, extensibility etc.	2 marks (3 mins)
Individual Part	
Individual members explaining their role/ contribution and any challenges	1 marks (4 x 1 min)

3. Face to Face Tutor Marking (Total 20 marks)

This final assessment point will award marks for both group and individual work with equal emphasis on individual and group contribution to design and coding. The group effort will be measured through overall class and use-case diagrams, which should explain the overall requirements and design decisions. Each individual member is expected to demonstrate their individual contribution by detailing their part of the use-case (a use case textual description) and class-diagrams and by coming up with two other UML diagrams (a sequence diagram and one of state/activity/object diagrams). **Tutors will check your diagrams are aligned with your code. Include a readme file identifying percentage contribution of each member allowing the group mark to be adjusted accordingly.**

Activity	**Marks**
Group Part	**Total 10 marks**
Overall Class Diagram (group) The class diagram must show all the methods, attributes, associations and multiplicities. Name the associations wherever possible.	Up to 2 marks for valid class diagram that captures all the main classes, associations (with naming) with the correct multiplicities. You are expected to follow the standard UML notation. The remaining 1 mark is allocated for consistency with individual sequence diagrams (you may have to update the class diagram to reflect the final sequence diagrams).
Overall Use Case Diagram (group)	Up to 2 marks for a valid Use case diagram capturing all individual use cases. For the remaining 1 mark show all the relationships including extends/includes.
Group Evaluation (max 1 page report) including: Lessons learnt about project/process management (2 marks) Strengths and weaknesses of your system in terms of reusability, extensibility, maintainability and usability (2 marks).	2+2 = 4 marks
Individual Part	**Total 10 marks**
Show detailed methods/attributes/associations relating to individual contribution in the class diagram	2 marks
Show detailed explanation relating to individual contribution in use case diagrams – present one use case textual description for a significant use case.	2 marks

Sequence Diagram (individual)	2 marks
One of state/activity/object diagram	2 marks
Code explanation relating to individual contribution. The code is expected to reflect the design. Code quality and explanation (code must be refactored)	2 marks

Submission Details: 31st May 2019 4.30 pm. No late submission allowed.

All source code and diagrams must be printed and bound together and be submitted to the School of engineering office (10.09.03) by 4.30 pm on the last day of the semester (31st May 2019). You must also upload a zip file called FinalSubmission.zip to your groups Trello board. Face to face marking sessions will commence on 3nd June.

Bonus Features should not to be attempted until the completion of Standard Features

(Up to 5 marks)

Notes: Bonus marks are applicable only to those individuals contributing to these features – who may be asked to explain the design rationale. Total mark for Assignment cannot exceed 40/40 (including progress and bonus marks)