

Software Engineering with Objects and Components  
 Group Tutorial Project: Deliverable 2  
*Marking Scheme*

**Part 1 - Team Information**

<b>Tutorial Group (T01-T06)</b>		<b>Tutor's Name</b>	
<b>Tutorial Group's Name</b>		<b>Team: S, T or A</b>	
<b>Team Members</b>			

**Part 2 - Individual Marking Scheme**

Individual Marking Scheme								
Team Member	Estimated Contribution				Individual Mark			
	[tick one box only]				[Basic + Bonus + Team Bonus]			
	U	S	G	E	Basic	Bonus	Team Bonus	Total

**U - Unsatisfactory:** Some attempt but,contribution very weak due to poor skills, unsatisfactory attendance or other factors. **Note that an unsatisfactory evaluation will result in reduced marks.**

**S - Satisfactory:** Met all team requirements and contributed ordinary.

**G - Good:** Above average effort and contribution.

**E - Excellent:** A sustained and outstanding contribution to team effort.

## Part 3 - Deliverable Marking Scheme

Deliverable Marking Scheme		
Deliverable Part	Questions	Marks
Prioritized Use Cases	Q1. Do your use cases capture functional system requirements? Do your use cases make good use of the features (e.g., use case relationships) of UML intended to support requirements gathering?	[ / 2 ]
Refined Class Diagram	Q2. Does your class diagram make good use of the features (e.g., attributes, methods and constraints that are necessary) of UML intended to support detailed design?	[ / 2 ]
Change History	Q3. Does your change history record change rationale? Does your change history enable (requirements) traceability practice?	[ / 6 ]
<b>Marks Limit: [10/100]</b>		[ / 10 ]
Sequence or Communication Diagrams	Q4. Are your sequence (communication) diagrams well-formed? Provide a brief assessment of how well your diagrams match the UML specification. Identify any instance of your diagrams deviating from the standard approach.	[ / 4 ]
	Q5. Do your sequence (communication) diagrams capture/realize your chosen Use Cases? Provide a brief assessment of the strengths and weaknesses of your sequence (communication) diagrams as a means of formalising the activity in your Use Cases.	[ / 8 ]
	Q6. Assess the extent to which your sequence (communication) diagrams are consistent with the class diagram.	[ / 8 ]
<b>Marks Limit: [20/100]</b>		[ / 20 ]
Activity or Statechart Diagrams	Q7. Are your activity (statechart) diagrams well-formed? Provide a brief assessment of how well your diagrams match the UML specification. Identify any instance of your diagrams deviating from the standard approach.	[ / 4 ]
	Q8. Do your activity (statechart) diagrams capture dynamic behaviours of your chosen/implemented classes? Provide a brief assessment of the strengths and weaknesses of your activity (statechart) diagrams as a means of formalising the dynamic aspects of your Use Case/Class Diagrams.	[ / 8 ]
	Q9. Assess the extent to which your activity (statechart) diagrams are consistent with the class model.	[ / 8 ]
<b>Marks Limit: [20/100]</b>		[ / 20 ]
Java Code	Q10. Do your Java classes make good use of Object-Oriented features (e.g., inheritance, visibility, etc.) intended to support Object Oriented design, team work and reuse?	[ / 4 ]
	Q11. Assess the extent to which your Java classes are consistent with the class diagram.	[ / 4 ]
	Q12. Do your Java classes compile and appear to function correctly?	[ / 4 ]
Unit Testing	Q13. Does the code built to implement your tests compile and test the system in the manner you intended in your test plan. Provide a justification for all the tests you want to apply to your chosen classes.	[ / 6 ]
	Q14. Do your classes pass all the tests? Do you believe your test code tests the system adequately? If Java classes fail to pass some testes, you should explain the problem and whether or not you have fixed it.	[ / 6 ]
	Q15. Assess how effective your tests are in detecting coding errors in your chosen classes. Provide a justification in term of the coverage you are achieving in the tests.	[ / 6 ]
	Q16. How extensible is your test code? If you decided to increase the number of tests how much extra effort would it take to incorporate those tests. Award higher marks for greater extensibility. Justify your award of mark by making a case for the extensibility of the test set.	[ / 6 ]
Integration Testing	Q17. Can you justify that you have addressed all the issues of integrating your classes in your system?	[ / 4 ]
	Q18. Have you addressed integration testing for the different parts of the system?	[ / 4 ]
	Q19. Can you justify that you have addressed all the most critical aspects of the integration of the work of the three teams?	[ / 6 ]
<b>Marks Limit: [50/100]</b>		[ / 50 ]
<b>Deliverable Mark</b>		[ /100]

## Deliverable 2 Marking Scheme - Instructions

This form is to be used by your team to assess your deliverable 2 and to distribute the mark among your team's individual members. *Please complete and return the form together with your deliverable 2.* This form consists of three parts.

**Part 1 - Team Information.** This part records your team's details. Please fill in the table in the Part 1 with the relevant information.

**Part 2 - Individual Marking Scheme.** This part records your team's distribution of the mark among individual team members. The final mark for an individual is the sum of three marks: [Basic + Bonus + Team Bonus]. Basic marks are calculated by a fixed calculation. Bonus marks are only available to individuals and teams who provided a distinguishable contribution. Bonus marks are to be determined **democratically** by the team. **Both, basic and bonus, marks have to be justified and accompanied by a supporting cases.** *Note that (i) no final mark of greater than 100 will be accepted; and (ii) marks will be checked and adjusted if necessary.*

**Basic Mark:** The basic mark for each individual is equal to the **Deliverable Mark** (see Part 3).

**Bonus Mark:** The **Bonus Mark of 5 marks** is available for distribution. Your team should decide, democratically - preferably amicably - whether this bonus mark should be given to **one member** of your team, who provided a distinguishable contribution towards your deliverable. **Note that, if you decide to allocate the bonus mark, you have to provide a supporting statement (involving any relevant technical or individual aspects) about the motivation of your decision.** Otherwise, the bonus mark is 0 for all team members.

**Team Bonus Mark:** The **Team Bonus Mark of 2 marks** is available for distribution. Your team should decide whether to give the team bonus mark to each individual by assessing the overall team work (e.g., performance, collaboration, tack and effort distribution, communication, etc.). **Note that, if you decide to allocate the bonus mark, you have to provide a supporting statement (involving any relevant technical or individual aspects) about the motivation of your decision.** Otherwise, the bonus mark is 0 for all team members.

**Part 3 - Deliverable Marking Scheme.** This part provides you a marking scheme for your deliverable 2. Your deliverable 2 should consist of different parts: *Prioritized Use Cases, Refined Class Diagram, Change History, Sequence (Communication) Diagrams, Activity (Statechart) Diagrams, Java Implementation and Testing.* The Deliverable Marking Scheme in Part 3 shows the maximum mark allocated to each part. You have to assess each part of your deliverable 2. The table provides relevant questions that structure your assessment process. You should assign (part of) the allocated marks for each question. The marks depend on the quality of your work. **Note that you have to justify any allocated mark by answering each corresponding question and providing an assessment supporting your mark allocation.** *You have to justify your mark assignment by answering the questions and proving an assessment in a separate sheet.* The **Deliverable Mark** is the sum of the marks of each part.