

RUBRIC FOR THE JAVA IMPLEMENTATION AND JUNIT TESTING ASSIGNMENT								
Criteria / Objective	Performance Level							
	7	6	5	4	3	2	1	Mark
Ability to implement complex application logic from a specification in a modern OO language, and to integrate with existing code. (7 marks in total)	The model classes are complete, professionally structured and pass both the student-supplied and assessment unit tests without error. The GUI based game runs without error. (7 marks)	Both classes are complete and well structured, but the testing process yields a number of errors which may affect game play occasionally. The GUI based game runs without apparent error for most cases. (6 marks)	Both classes are complete, but errors are found during testing which may affect routine game play. The GUI usually runs without error. (5 marks)	Some aspects of one or more model classes are poorly structured or plainly incorrect. The GUI exhibits some instability when the code is integrated. The unit testing process exhibits many errors. (3-4 marks)	The model classes are markedly incomplete or non-existent. (0-2 marks)	/7		
Ability to design and implement professional unit test coverage from a specification under a modern unit testing framework. (8 marks in total)	The testing strategy and test implementation fully exercises the range of operations specified. Each non-trivial method has several specified test cases, and the supplied unit tests capture all or almost all of the variations in the 'broken' test code. The unit test methods are well designed and take advantage of the facilities provided by the test harness. (7-8 marks)	The strategy exercises the range of operations but the test cases are less complete, with for example some obvious boundary or initial cases not considered. The unit test methods are well designed and take advantage of the facilities of the test harness, but the supplied unit tests fail to identify some of the problems in the broken test code. (6 marks)	Coverage remains good, but is less complete than at the higher levels. Some methods perhaps have inadequate coverage, but the overall strategy is good. The use of the framework is reasonable, but perhaps could be better structured. (5 marks)	Coverage is reasonable but there are significant omissions. Some methods are poorly covered, and the overall impression is of arbitrary test cases for the sake of completeness rather than as part of careful planning. The broken test code reveals many flaws in the strategy. (3-4 marks)	Coverage is weak and in places almost non-existent. There are a number of methods which are barely tested at all. The use of the JUnit framework is poor and the supplied tests fail to capture much of the aberrant behaviour of the broken test code (0-1 marks)	/8		
Ability to write code and documentation to a professional standard. (5 marks in total)	Both the java model and test code conforms to the standard presented in the supplied classes, interfaces and javadoc. Identifiers are self-documenting, commenting (including javadoc) is consistent, light and aids comprehension. There is appropriate use of whitespace, and indentation is used to enhance understanding of structure. (5 marks)	Both the java model and test code basically conform to the standard, although there are some minor inconsistencies, excesses of commenting or poorly chosen identifiers. Identifiers are self-documenting, commenting is consistent, light and aids comprehension. Indentation and whitespace are usually consistent and an aid to comprehension. (3-4 marks)		There is some attempt to employ good commenting, appropriate use of whitespace and self-documenting identifiers but this is often not consistent with the coding standard or itself inconsistently applied. Overall effect does not appear professional. (2 marks)	Documentation is weak or non-existent, identifiers are often poorly chosen and the use of whitespace and indentation is poorly considered. Overall impression is very disorganised and unprofessional. (0-1 marks)	/5		

/20

Additional comments: