Two hours

Question ONE is COMPULSORY

Note: throughout this exam, the phrase “MELT project” refers to the project as a whole as discussed in the course, not just the part you implemented in the lab. Where the latter is meant, a phrase such as “your MELT implementation” is used.

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Software Engineering Overview

Date: Wednesday 21st January 2015
Time: 09:45 - 11:45

Please answer Question 1 and two other Questions

This is a CLOSED book examination

The use of electronic calculators is NOT permitted
Question 1

This question is **COMPULSORY**

a) The Waterfall method of software development is now obsolete. State the main reason why it failed, and one other. 

(2 marks)

b) Give two examples of applications, *not* discussed in the course, where the amount of time spent on ceremony is likely to be more than 20% of the total for the project.

(2 marks)

c) Briefly define the “boss test” and give an example *not* given in the course of a use case which would pass it.

(2 marks)

d) State what kinds of things appear on a design class diagram but don’t appear on a domain class diagram, and one other way in which the two kinds of diagrams differ.

(2 marks)

e) Give two examples of classes in your MELT implementation, one of which is a Pure Fabrication and one which is not.

(2 marks)

f) Suppose you are implementing a CAD system for designing large buildings. Such a building will have a very large number of components, which will need to be represented as objects. Give two reasons why it would probably be a good idea to create these objects from a factory.

(2 marks)

g) Give one advantage and one limitation of unit testing Java code with JUnit.

(2 marks)

h) A successful test is one which causes the software to fail, but this is difficult for programmers to do with code they have written. State two different approaches to testing which aim to get around this problem.

(2 marks)

i) Briefly state the concept underlying Design by Contract and how it relates to formal methods of software development.

(2 marks)

j) Give one advantage, and one potential disadvantage, of aspects as used in AspectJ.

(2 marks)
Question 2

a) The four key principles of the Agile Manifesto can be considered *ethical* principles. For each, state how this is the case. (4 marks)

b) State four different groups of stakeholders, other than students, in a real-life version of the MELT project, and what their principal concerns will be. (4 marks)

c) Give six rules (things to do or not do) in writing a UC, using the Take Test use case for MELT for examples as. (6 marks)

d) For each of the 6 types of requirement in the FURPS+ checklist, briefly state what they are, and for each give an example for a Computer Aided Design system used to design civil aircraft. (6 marks)

Question 3

a) How can UML class diagrams be used for communication *between* software developers? Your answer should mention at least two different uses. (4 marks)

b) Explain, using an example from MELT, how a domain class diagram can be used to gather useful information from stakeholders. Your answer should take into account the different kinds of skills which different stakeholders have. (5 marks)

c) Draw a *domain* class diagram which shows the following information about the representation of rooms in a university-wide timetabling system:

“There are three kinds of room to be considered: standard teaching rooms, which can be used for any class which does not require specialist equipment, computer labs, used for all classes requiring computers, and specialist rooms, containing specialist equipment. In the last case, the kind(s) of specialist equipment are recorded, but not in detail because it is assumed that these rooms will only be used for classes in the school which contains them, e.g. only Chemistry students will use Chemistry labs. For every room, the school containing it, and the location within the school is also recorded. Also, it’s important to know the capacity (number of seats) of every room.”

Hint: do **not** add information not explicit in the above description. (5 marks)

(Question 3 continues on the following page)
(Question 3 continues from the previous page)

d) State the fundamental rule which determines whether a use of inheritance is appropriate, and briefly explain why it is important to obey this rule. (2 marks)

e) Give an example of inheritance not given in the course, which does obey the rule, and another which does not. (2 marks)

f) Java has a very different approach to inheritance compared to other common OO languages, with only single inheritance (with the exception of interfaces, which have very limited functionality). Common “mixin” cases are dealt with by having some functionality (e.g. synchronization) built in. State whether in you view, this restricts the scope of a Java programmer in practice, briefly explaining why or why not. (2 marks)

Question 4

a) Briefly explain what we should always aim for in terms of Coupling and Cohesion and give an example of each from your MELT implementation. (3 marks)

b) Explain the GRASP principles of Polymorphism and Protected Variations, and how they are related, using examples from MELT. (6 marks)

c) Explain the notion of a Controller, and the different types of controller, using an example of each kind. (3 marks)

d) Suggest whether the use of controllers is appropriate for use in a MELT implementation, briefly explaining why or why not. (2 marks)

e) Explain the notion of a Factory, and how wise use of factories can have advantages in terms of GRASP principles. (4 marks)

f) Under what circumstances is it appropriate to sacrifice adherence to GRASP principles for advantages in performance (e.g. shorter runtime or lower space usage). (2 marks)
Question 5

a) What is System Testing, as defined in the course? (1 mark)

b) Give four examples of things which would need to be done in system testing for MELT, including at least two which don’t directly involve testing the code. (4 marks)

c) What happens if, in the course of system testing, new requirements emerge, and what does this imply about the timing of system testing? (2 marks)

d) Embury’s Law states “This software has bugs in, we just don’t know what they are yet”, where “this software” is any system of any significant size. Explain why this is almost always true, and give one possible counterexample. (4 marks)

e) State four things that can be done in a project to minimise the impact of Embury’s Law, beyond putting a lot of effort into testing generally. (4 marks)

f) Estimate how many bugs there are left in your MELT application. You should combine the estimates of bug density given in the lectures, with your practical experience of the project, to come up with a realistic estimate. (5 marks)