Two hours

Question ONE is COMPULSORY

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Software Design using Patterns

Date: Thursday 15th 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) “Design patterns are a tool for communication”. Explain this statement, using an example. Your explanation should state who is doing the communication and what is being communicated. (4 marks)

b) Constructo Ltd. make custom toys. They have a range of basic parts (e.g. wheels, cogs, bricks etc.), which are made up into a wide range of different toys, often containing common sub-parts (e.g. a wheel-axle-gearbox combination is common to a number of toy cars). Toys are shipped with just the required parts and assembly instructions, but first each toy must be assembled at the factory to ensure it all fits together nicely. Give a UML diagram which shows how the Composite design pattern can be used to represent these toys. Note: you do not need to represent the different kinds of basic parts. (4 marks)

c) Briefly explain the advantages of using the Composite pattern rather than having a simple list of parts in the scenario of part b. (2 marks)

d) Briefly explain the advantages of using the Composite pattern rather than having a simple list in a different scenario than that of part b. (2 marks)

e) Constructo Ltd is a large company and so, at any one time, they are in the process of shipping many toys, by a variety of forms of transport (e.g. vans, lorries, containers). They need to keep track of all of these so, for instance, company managers can see what’s happening and optimise their schedules, and customers can check the status of their orders (as happens with, e.g. Amazon). Draw a UML diagram which shows how the Observer pattern can be used here. (5 marks)

f) Observer is often represented as a two-step process, but actually there are three steps involved. Briefly explain what these three steps are, in the context of the example of part e). (3 marks)
Question 2

a) A slogan for agile software development is “embrace change”. One kind of change is requirements change caused by stakeholder feedback. What other kinds of change can happen (to requirements or otherwise) during a substantial software project? (4 marks)

b) Reminder: Simon Smartsuite is the University’s head of IT, who, while fundamentally on your side, has some very old-fashioned ideas about software process – he still believes in Waterfall!

Simon has “got the right result” with the University’s Senior management team, so the project to build Computer Assisted Assessment software for the University is on. You decide that it’s time to educate him about the UP, and that it’s hopeless trying to do this via email, so instead you take him out to dinner to thank him for his work with the SMT and to try to explain the advantages of the UP. Give six points you would make about the agile UP, without ever naming the phases of the UP. (6 marks)

c) The University of Mancunia decides to implement a University-wide web information system, to go on line for the start of the next academic year, in 9 months time. This will contain a large amount of information, including the web page of every member of academic staff, and a great deal of information for students.

Currently, web pages are held in various different formats in different schools, and over 200 academics and admin staff are involved. Under the new system everybody will use the same software and the information will be stored in a central database which will form the “single point of truth” for all web information in the University.

What will be the main risks associated with this development, as perceived by the following groups of stakeholders?

i) Students (2 marks)

ii) IT support staff (1 mark)

iii) University management (1 mark)

iv) Admin staff responsible for maintaining web pages (hint: consider both short-term and long-term risks from their perspective). (2 marks)

d) Of the four groups of stakeholders mentioned above, state for each how important it would be to have a representative interacting frequently with the development team, briefly saying why in each case. (4 marks)
Question 3

a) It’s often said that design patterns are a tool for communication. Explain with an example *not* given in the course, why the same is true for GRASP principles. (4 marks)

b) The Information Expert and Creator principles have the common property that they can both be used “in reverse”. Briefly explain each of these principles, and how they are used in reverse. (4 marks)

c) “Protected Variations is actually an antipattern”. Explain what is meant by this statement, and why it might actually be true in some circumstances. (4 marks)

d) Explain how Polymorphism often gives us Protected Variations in a way which is actually useful. (2 marks)

e) In the early days of C++, inheritance, and in particular multiple inheritance, was often overused. It was common for a class to have multiple superclasses, with no is-a-kind-of relationship between superclasses and subclasses. Explain, in terms of GRASP principles, why such a design is very bad. (4 marks)

f) The Java designers took a very different approach, with only single inheritance, with the exception of interfaces, which have very limited functionality. Common “mixin” cases were 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) There are three design patterns which specifically address the problem of representing complex algorithms in OO. Briefly explain how the problem arises, and how each of these patterns can help. (4 marks)

b) Briefly explain, with an example, why there are many cases where the problem mentioned in part a) simply does not arise. (2 marks)

c) Explain the idea behind the Flyweight pattern, and why it is different from many design patterns. (3 marks)

d) At the time the GoF book was written, Flyweight was a commonly used pattern. Nowadays, it is much less often used – explain why. (3 marks)

e) The GoF book lists a number of conditions which have to apply for the Flyweight pattern to be applicable. One is “The application doesn’t depend on object identity”. Why would this be a problem with the use of Flyweights? (2 marks)

f) Suppose you have an engineering application which creates many parts. Sometimes there is no problem with doing it the simple way, but at other times there are so many parts that you are forced to use Flyweight. You may not know which situation you are in until runtime. Draw a UML diagram which shows how the Strategy pattern can help in this situation. (4 marks)

g) Why is switching from one strategy to another more difficult in this case than the chess playing example given in the lectures? (2 marks)
Question 5

a) Explain, using an example other than that in part b, the problem which Aspect-Oriented Programming is designed to solve. (4 marks)

b) Briefly explain what the two main constructs within an aspect in AspectJ are, giving an example of how they could be used to deal with the need to lock certain objects before they are used in a multithreaded application. Accurate AspectJ syntax is not required, provided the meaning is clear. (6 marks)

c) The use of AOP has been advocated in a large number of different situations. For each of the following, state how plausible the claim is that AOP improves the design, briefly explaining why.

   i) In the implementation of Computer Assisted Assessment software (2 marks)

   ii) In the use of Design by Contract. (2 marks)

   iii) In the implementation of design patterns. (2 marks)

d) State the advantages and disadvantages of AOP which should be taken into account when deciding whether or not to use it on a project. (4 marks)