COMP23420

One and a half hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Software Engineering

Date: Thursday 16th May 2013

Time: 09:45 - 11:15

Answer Question ONE and also either Question 2 or Question 3

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

Question 1

This question is COMPULSORY

a) Given the following two rules:

1. "A driver may specify up to two resting days for each week in which they will not be available for work."
2. "A driver can drive for a maximum of 5 hours at any one time and must have a break of at least one hour."

State which rule is for driver scheduling and which one is for driver rostering.

Explain the purpose of driver scheduling and driver rostering and the relationship between them.

(4 marks)

b) Briefly explain why driver scheduling is a hard problem and the methods for solving this problem.

(3 marks)

c) Explain the principles of data abstraction, information hiding and encapsulation in Object-Oriented Programming.

(3 marks)

d) An important principle in agile software development is *Physicality* - physical interactions are considered better than virtual interactions. Explain why this is.

(2 marks)

e) Give three examples of agile practices which are consistent with the principle of Physicality, and for each state whether your team did it or not, and why.

(3 marks)

f). Why is it not a good idea for a programmer to write some code and then test it?

(1 mark)

g). State two alternative approaches to testing which overcome the problem of part f).

(2 marks)

h) Give an example *not* given in the lectures of a JUnit fixture. Hint: you need a JUnit annotation and a small piece of Java code. Minor errors in the code will not be penalised.

(2 marks)

Question 2

Answer this question OR question 3

a) Explain why pure classes are not enough to complete the design of a complex object-oriented system and how higher-level design constructs can overcome the limitations of pure classes. (5 marks)

b) Explain, through an example, how the Role pattern works. (5 marks)

c) Briefly explain the role of GRASP principles in object-oriented software development. (2 marks)

d) Explain how the Role pattern which you explained in part b is consistent with **five** different GRASP principles.

(5 marks)

e) Explain how your implementation of the roster and the controllers' user interface was consistent with the GRASP principles of Information Expert, low external Coupling and Protected Variations (or if it was not, explain why not).

(3 marks)

[PTO]

Question 3

Answer this question OR question 2

You are designing a point-of-sale software system for a convenience store. You want to apply some of the business application patterns you have learned from this course to your design.

a) Design a UML class diagram to show the relationships between the following three business domain objects: Sale, Product and Line Product. Your diagram should show: (1) the multiplicities of these relationships; (2) essential attributes of these objects and essential operations performed by these objects.

(4 marks)

b) Explain the business application design patterns used in your design.

(2 marks)

c) Extend your design by showing cash and card payment and explain new patterns used in this extension.

(4 marks)

d) Other than sheer size, what are the main characteristics of a piece of software which affect how easy or difficult it is to test? Illustrate your answer by briefly explaining how difficult or otherwise it would be to test (1) the subsystem you designed for part c, and (2) the whole sales system of which it is a part.

(5 marks).

e) Estimate how many bugs there are likely to be left in your IBMS implementation. Explain your reasoning, take into account the estimates of bug density given in the lectures and the factors relevant to bug density relevant for your project.

(5 marks)

END OF EXAMINATION