One & half hours - on line

The exam will be taken on line.
This is the paper format, which will be available as a backup
and to be handed out to students for reference immediately AFTER the examination starts

Please do NOT use the exam paper to write your answers

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Software Engineering

Date: Thursday 19th May 2011

Time: 14:00 - 15:30

Answer Section A
and EITHER Question B1 OR Question B2

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

Section A
Answer All Questions (Short Questions)

a) What is the difference between a domain object model (also called “a domain model”) and a design class model (also called “a design class diagram”)? What is contained within a domain glossary?

(2 marks)

b) Object-oriented development processes can be divided into a macro development process and a micro development process. Briefly describe the purposes of these two processes and their relationship.

(2 marks)

c) What is the purpose of object-oriented analysis (OOA)? What is the purpose of object-oriented design (OOD)? According to your IBMS project, what are the three artefacts from OOA?

(2 marks)

d) What is the name of the software architecture used in your IBMS project? Provide a general description of its constituent parts and then relate these parts to the specific components in your project.

(2 marks)

e) What is the purpose of requirements elicitation? What is the main difference between functional requirements and non-functional requirements?

(2 marks)

f) Consider the following design classes for the part of a restaurant management system concerned with staff members:

```
public abstract class StaffMember

public class Waiter extends StaffMember
public class Chef extends StaffMember
public class Manager extends StaffMember
```

Assuming that it is necessary to store information about staff members, what is the potential problem with this design?

(2 marks)

g). Briefly explain how a different design can avoid this problem and suggest another advantage of this alternative design. (2 marks)

(Question 1 continues on the following page)

(Question 1 continues from the previous page)

h) In the naïve approach to testing a programmer writes the code and then tests it. Briefly explain two other approaches and their advantages over the naïve approach. (2 marks)

i) Briefly explain, using an example not given in the course, the difference between *Integration Testing* and *System Testing*, and why it matters. (2 marks)

j) "Physicality" is considered important in agile software development. Give two specific examples of agile practices that promote physicality. (2 marks)

Section B

Please answer ONE question

B1. Figure 1 below depicts an Internet-based flight booking system. The system provides 9 use cases for its passengers. The system is supported by 5 external systems and services. According to Figure 1, answer the following questions:

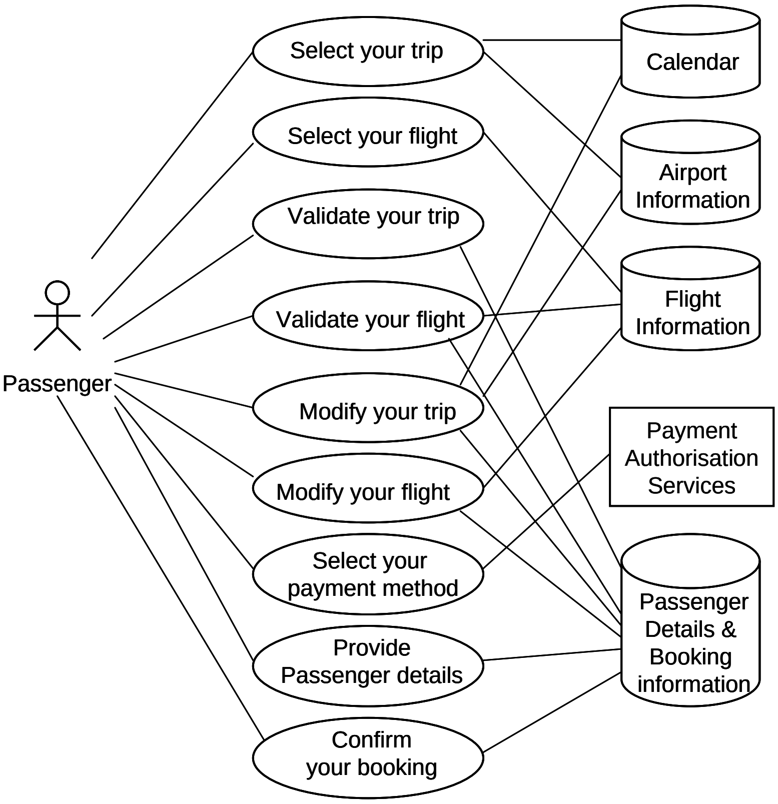

Figure 1: An Internet-based flight booking system

a) How many actors participate in this system? Name these actors and concisely describe the role of each actor in this system. Which actor is the primary actor?

(5 marks)

b) Provide a detailed use case specification for “Select your trip” and include in your specification the precondition and the post-condition. Assume that you are selecting a round trip.

(8 marks)

(Question B1 continues on the following page)

(Question B1 continues from the previous page)

c) Name the domain objects (terms) identified in your use case specification.
(2 marks)

d) Based on your use case specification and your choice of domain objects, draw a design class diagram. Concisely justify your choice of design classes. For each class, show its main attributes and operations.
(5 marks)

B2

a). Explain the GRASP principles of High Cohesion and Low Coupling and how they are related. Your answer should distinguish between two different kinds of coupling. (5 marks)

b). Constructor Ltd makes construction kits for children. Kits consist of parts of various different kinds. They can be broadly classified as Simple parts, such as cogs and levers, and Compound parts which are pre-built from simple parts. Simple parts have properties such as colour and, of course, cost. Your team is building a software system to enable the company to build and sell kits more efficiently.

i) A junior software developer joins your team. You don't know anything about him so you set him a test – you ask him to come up with design classes corresponding to the description of parts etc. above. He comes back with a design that contains a single class, called Parts. He explains:

"I used just one class so it's nice and cohesive, and to minimise coupling. I also found an opportunity for code reuse, so here I have a list that contains parts for a kit, or simple parts for a compound part or features for a simple part. This instance variable TYPE tells you which type of things a particular instance represents."

Explain in terms of the GRASP principles of High Cohesion, Low Coupling and Protected Variations why this is a very bad design. (3 marks)

ii) Draw a class diagram that shows a better design based on the GRASP principle of Polymorphism. (5 marks)

iii) Explain how this use of Polymorphism yields a design that conforms to several other GRASP principles. (4 marks)

iv) Explain how a method could be implemented to calculate the total cost of a Kit from the costs of the parts it contains. You do not need to write the code, just explain clearly.
(3 marks.)

END OF EXAMINATION