H3 – CONTEMPORARY ISSUES IN SYSTEMS DESIGN
SOLUTIONS & MARKING SCHEME June 2013

Part A

The purpose of these questions is to establish that the students understand the basic ideas that underpin the course. The answers should be largely descriptive and quite short.

Answer A1.

Identify five underlying causes of problems in information systems development.

Addresses L01. The course text lists the following (in Chapter 2) the wrong problem is addressed, wider issues are ignored, incorrect analysis, the project is started for the wrong reason, users change their minds, changes in the environment, the implementation is not feasible, poor control by managers. Any five of these would be appropriate with 1 mark for each identified.

Answer A2.

What does requirements traceability mean in object oriented development?

Addresses L02. The term requirements traceability refers to the capability of tracking each requirement to all the related systems development deliverables (2), from analysis models (e.g. from use cases to sequence diagrams(1) to class diagrams(1)) to program code (1).

Answer A3.

What is the purpose of a system repository?

Addresses L03. A repository is part of a CASE tool (1) and should hold the descriptions and specifications of all modelling elements. The class diagram will be particularly important (1) so the repository should include information on (1), attributes (1) and operations (1).

Answer A4.

How does the object-oriented concept of message passing help to hide the implementation of an object including its data?

Addresses L05. Other parts of a system only see an object's interface (1) (services it can perform and operation signatures (1)). Internal details including data are hidden (1) and can only be accessed by a message (1) that contains a valid signature (1).

Answer A5.

What is the main difference between USDP and the waterfall lifecycle in the relationships between activities and phases?

Addresses L03. In the waterfall life cycle, activities and phases are effectively the same (1), e.g. analysis activities take place in the analysis phase or stage (1). In an iterative life cycle like the USDP, the same activities represented by workflows (1) take place in each phase (1), but the balance of activities changes as the life cycle progresses (1).

Answer A6.

Outline the main steps in developing a class diagram from a use case.

Addresses L04. The main approach followed in Chapter 7 of the course text has these steps: From the use case description, identify possible classes (1); Identify a possible sequence of messages that could achieve the objectives of the use case (1); Draw a communication diagram to represent this interaction; Review the communication diagram from an architectural point of view, e.g. by adding boundary and control objects (1); Review the requirements of the use case, adding other links and messages as necessary(1); Convert the communication diagram to a class diagram by changing objects to classes and links to associations (1).

Answer A7.

What does the term pattern mean in the context of software development?

Addresses L04. A definition of pattern is given in Chapter 8 of the course text. According to this definition a pattern comprises three elements: a context in which a given problem occurs repeatedly (2), a set of forces that influence or constrain the possible solutions (2) and a software configuration that allows these forces to be resolved (1).

Answer A8.

Identify two specific features of bad object oriented modelling that are discouraged by the use of communication diagrams.

Addresses L04. Communication diagrams discourage both using a large number of messages between two objects (2) and having too many parameters for each message (2) as these are clumsy to represent on the diagram (1).

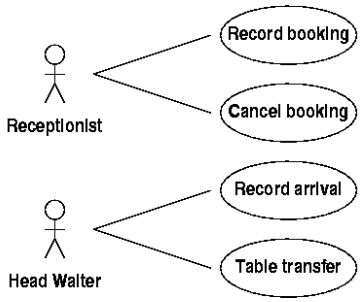
PART B

Answer B9. - Addresses L03.

You have been asked to develop a booking system for a restaurant. The following key terms have been defined by the restaurant owners:
Booking: an assignment of diners to a table
Covers: the number of diners for a booking
Customer: a person who makes a reservation
Reservation: a booking made in advance
Walk-in: a booking that is not made in advance

Identify four Use Cases for the booking system and briefly describe two of them. (5 marks)

The following are four most obvious Use Cases (1):

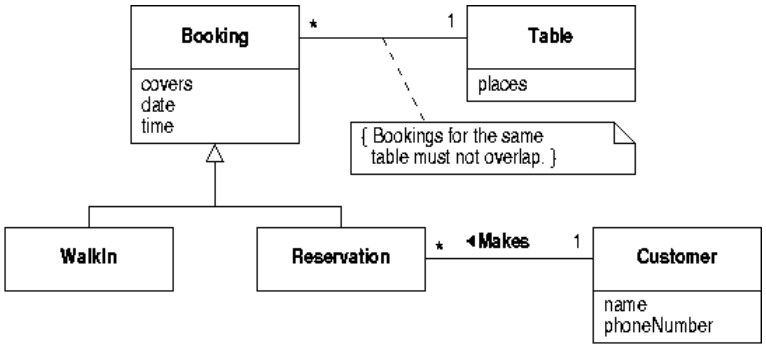


The two selected Use Cases should be described to the following level of detail (2 marks for each description):

Record Booking:

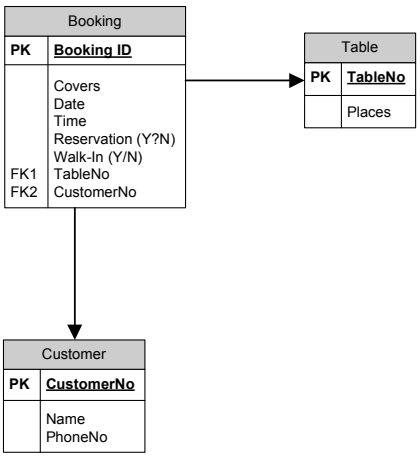
Receptionist enters required date.
System displays bookings for that date.
Suitable table identified – receptionist records customer name and phone no. along with table number and number of covers.

Create a class diagram for the booking system from the above information. Include any attributes and methods that may be needed. (10 marks)



5 marks for correctly identifying the classes, 3 for the relationships 2 for any attributes/methods.

Create a data model showing how data in your class diagram could be stored in a relational database. Your data model should include any primary and foreign keys that would be needed. (5 marks)



3 marks for correct use of primary and foreign keys; 2 for converting inheritance relationship to a single table.

Answer B10. - Addresses L04.

Define and provide an example to illustrate the following key concepts in object oriented development:
Classes, objects attributes and methods (4 marks)

Class is a general template (1) which is used to define objects (which is an 'instance' of a class) (1). Objects have attributes. For example: you have a class "people" and an object "students" which is derived from "people". The object 'students' has attributes like student-last-name; student-first-name; student-ID, student-phone, student-major, etc. (1). A method is nothing more than an action that an object can perform – like a function or procedure in a traditional programming language (1).

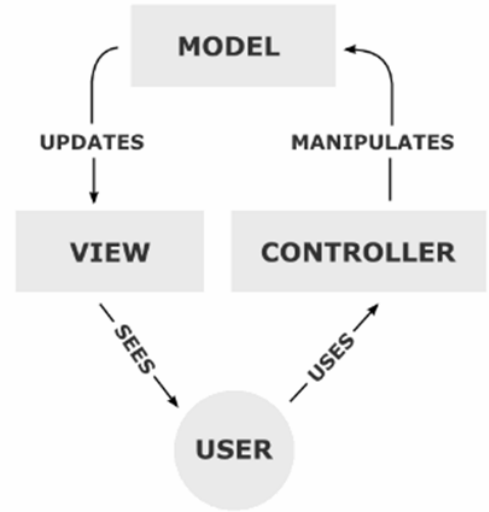
Inheritance and polymorphism (4 marks)

Inheritance is that a subclass 'inherits' common sets of attributes and methods from a class above it. For example, 'employee' might be an abstract class and be made up of sales-person, manager, buyer, etc. Each of the subclasses would inherit the attributes and methods from the class above it (such as name, employee-id, employee phone number, employee-address and more) (2 marks).
Polymorphism means that the same message can be interpreted differently by different classes of objects. For example "insert" – inserting a new patient into a medical appointment system and inserting a new item into an inventory system both use 'insert', but the object will interpret the 'insert' action differently depending on the different class of objects (2 marks).

What are the four fundamental Unified Modelling Language (UML) diagrams? And in which order do you generally see them developed? (4 marks)

Use Case Diagrams; class diagrams; sequence diagrams and behavioural state machine diagrams (2 marks). Use Cases always are created first; and behavioural state machines are almost always created last and the two in the middle (class diagrams and sequence diagrams) can be in either order, depending upon the project and the preferences of the analyst(s) (2 marks).

Many object oriented systems use the Model View Controller (MVC) architecture. Describe the structure of MVC and explain how it is commonly used (8 marks).
The following diagram presents the three components of MVC and their interrelationships (2):



The key components are:
*A **controller** can send commands to its associated view to change the view's presentation of the model (e.g., by scrolling through a document). It can also send commands to the model to update the model's state (e.g., editing a document). (2)*

*A **model** notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A passive implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them. (2)*
*A **view** requests from the model the information that it needs to generate an output representation to the user. (2)*
The pattern is used widely in web development and underpins Microsoft's document/view architecture.

Answer B11. - Addresses L02.

Identify five general advantages that might be gained from using a methodology during systems development (5 marks)

The course text identifies the following (1 mark for each):
Better quality system product
More complete satisfaction of user requirements
Greater management control
Better communication among developers and with users
Capture of know-how and its dissemination through the organization

What is the main difference between the "hard" and "soft" systems view of system development? When should each be used? (4 marks)

The hard systems approach is based on an assumption that the reality of systems development can be described in a way that is beyond reasonable dispute (1), and that this description can be subjected to rational analysis(1). Hard systems methodologies assume that system objectives can be clearly identified and defined (1). This can lead to solving the wrong problem or developing a wonderful system that nobody needs (1).

What is the difference between systems prototyping and throwaway prototyping methodologies? When should each be used (3 marks)

Systems prototyping generally leads to a functional system(1); while throwaway prototyping generally leads to understanding the user requirements (1) and design considerations more quickly (1).

Identify four criteria to be considered when selecting the most appropriate methodology for a particular project (8 marks)

Clarity of user requirements (2) – this would encourage prototyping and an agile approach.
System complexity (2) – this would encourage a model-driven, incremental approach.
Time schedules (2) - tight schedules would encourage timeboxing and prototyping.

Familiarity with technology (2) – throwaway prototyping would be particularly appropriate where programmers are working with new technology – encourages early experimentation with the new tools.

Answer B12. – Addresses L05

What three types of benefit might be gained from using reusable components in software development? (6 marks)

The arguments for reuse fall into three main categories. Two marks for discussion of each category:
Economic arguments: If some requirements can be satisfied using components that were developed on another project some time and money should be saved - though these savings will be slightly offset by the cost of organising and maintaining a catalogue of reusable components.
Quality arguments: If a component has been tested on another project less time need be spent with quality tests for the new project.
Business Flexibility: Organisations need to be able to respond to changes in a business environment by recombining information system components in new ways to create new business processes or adapt existing ones.

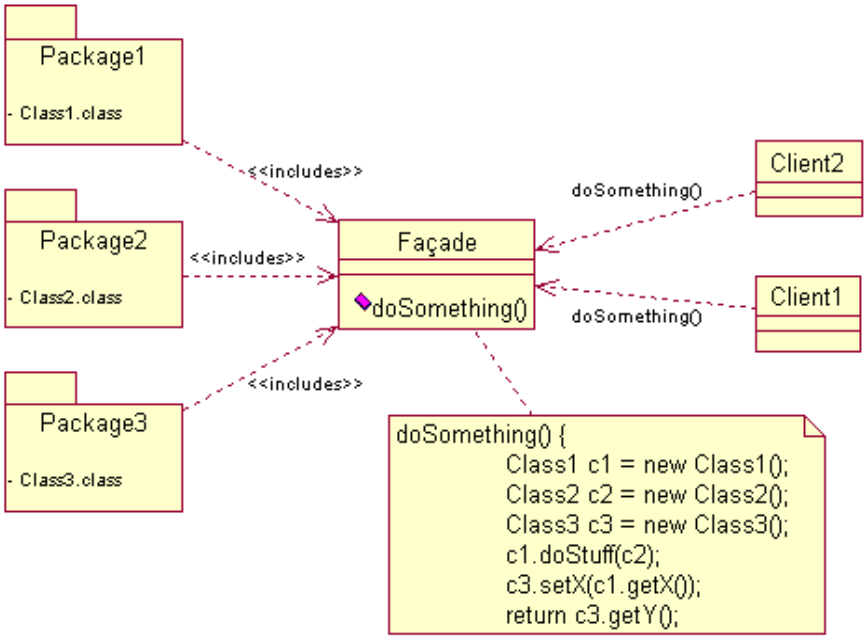
Object oriented systems have not achieved the level of reuse that was expected of them in terms of generating reusable components. Identify three technical reasons why this might have happened. (6 marks)

Lack of standards for reusable components (2 marks). This has been a problem but the introduction of standards such as the OMG's Reusable Asset Specification (RAS) might alleviate this in the future.
The platform dependence of reusable components (2 marks). Most components run on a particular platform – web services offer the prospect of a standard protocol for future developments.
High-coupling between different classes in object oriented design (2 marks). The class is often thought of a unit of reuse but classes are often tightly integrated with other classes in a particular application.

Describe the Façade pattern and explain how it encourages re-use (8 marks)

This pattern is described in some detail in the course text - in the chapter on Software Reuse (Chapter 20. The pattern describes how a simple interface can be created to a complex subsystem.

The structure of the pattern might be described in a diagram like this (3 marks)



The facade class above abstracts Packages 1, 2, and 3 from the rest of the application (1).
The objects are using the Facade Pattern to access resources from the Packages (1).

The pattern encourages reuse by providing a higher-level interface that makes the sub-system easier to use because the operations are all operations of a single Façade class (2). The developer using the sub-system does not need to know the details of the internal structure of the sub-system (1).

Answer B13. - Addresses L01

The failure of an Information Systems Project can be the result of “quality” problems.

Identify seven quality problems that may occur in systems development. Comment on the ways in which we can minimize the risk of each problem occurring. (12 marks)

The course text discusses this topic in Chapter 3. There the following quality problems are identified (2 marks for each identified):

Problem	How to minimise risks
Wrong problem is addressed	Systematic approach; business modelling; use case modelling.
Project undertaken for wrong reason	Business modelling, use case modelling, prototyping.
Wider influences are neglected.	Requirements prototyping
Missing or inappropriate functionality	Use case modelling, prototyping.
Incorrect requirement analysis	Use Case modelling, traceable development
Users change their minds	Agile development
External events change the environment	Systematic approach; business modelling.
Poor interface design	Prototyping
Inappropriate data entry	Prototyping
Software causes inappropriate ways of working	User involvement
Incomprehensible error messages	User involvement
Unhelpful “help”	User involvement
Requirements changed before project delivery	Agile development

Identify three different ways of involving users in systems development? What are the potential problems with each of these? (6 marks)

Users can be involved in the systems development process in the roles shown in the table below (2 marks for each).

Role	Problems
Full team member	Can lose sight of the user perspective but this can be overcome by rotating the team membership.
Consultative and Review	No direct influence on the design of the new system.
Participant in fact finding as interviewee only	Lacks sense of ownership for the new system.

A Questions	*LO1*	*LO2*	*LO3*	*LO4*	*LO5*
1	/				
2		/			
3			/		
4					/
5			/		
6				/	
7				/	
8				/	
B Questions					
1			/		
2				/	
3		/			
4					/
5	/				