IMIS Diploma

December 2013 Exam Paper Marking Scheme

Unit Title: H3 Contemporary Information Systems Design

Part A: Answer all of the following 8 questions (5 marks each)

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.

1. Identify **THREE** ways in which clients, users and developers might disagree about the most important objectives of a systems development project.

Addresses L01. The examinee should give three responses each worth two marks to a maximum of five marks. Suitable responses include: They differ in their view of the problems because their view of the meaning and purpose of IS development also differs; We might expect users to be concerned with the usability of the system or the impact it will have on the day to day nature of their work; clients will also have to be concerned with savings that might be made with the systemand the cost of developing it.

2. Identify **THREE** disadvantages of the waterfall model of systems development.

Addresses L02. The examinee should give three responses each worth two marks to a maximum of five marks. Suitable responses include:

- *real projects rarely follow a simple sequential life cycle.*
- *iterations are almost inevitable.*
- *the elapsed time between inception and delivery is frequently too long.*
- *it is unresponsive to changes in the technology or requirements.*

3. What is meant by "polymorphism" in the context of object oriented design?

Addresses L05. Polymorphism means that when one message is sent to objects of different types, each has an appropriate, but different, implementation for its response[3]. The object that sends the message need not know which type of object is addressed. One way of implementing polymorphism is through inheritance and overriding[2].

4. What is the **KEY** difference between "structural" and "behavioural" diagrams in the Unified Modelling Language?

Addresses L03. The following responses are a little more detailed than should be necessary to obtain full marks. The examiner should use a little discretion in deciding whether the examinee has shown that they have the basic ideas.

Behaviour diagrams emphasize what must happen in the system being modelled. Since behaviour diagrams illustrate the behaviour of a system, they are used extensively to describe the functionality of software systems[1]. Use Case diagrams, Statecharts, activity diagrams, sequence and communication diagrams are examples.[1]

Structure diagrams emphasize the things that must be present in the system being modelled. Since structure diagrams represent the structure, they are used extensively in documenting the software architecture of software systems.[1] Class diagrams, Component diagrams, Deployment diagrams and Package diagrams are examples[2].

5. Outline the **FIVE** main steps in developing a sequence diagram from a Use Case.

Addresses L03. First, identify the classes that interact with each other during the use case scenario[1]. List these across the top of the diagram[1]. Next, add arrows to represent the messages being passed from object to object[1]. Label the arrows with descriptive terms and include any parameters that are passed as a part of the message[1]. Draw lifelines under each class, and terminate those that cease to exist within the use case by placing an X at its end. Draw narrow rectangles down the lifeline to show when the classes are sending and receiving messages.[1]

6. Explain how object-oriented design is well-suited to the design of reusable components.

Addresses L05. OO encourages highly modular design; encapsulation and message passing encourages information hiding; inheritance and polymorphism encourage customisation [3]. OO fits well with component-based software development. [2]

7. Identify the **TWO** main purposes of an operation specification.

Addresses L04. They confirm that the logic of user requirements has been understood correctly and documented accurately[2]. They provide designers and programmers with a detailed and unambiguous basis for the design and implementation of system behaviour[3].

8. How can collection classes be used when implementing associations between classes? Illustrate your answer with a suitable example.

Addresses L03. The collection class can manage the collection of objects [2] at the "many" end of an association relationship. Example [3] for a constructive example.

Part B: Answer 3 of the following 5 questions (20 marks each)

Question B9 - Addresses L05.

a) Explain how software reuse might benefit an organisation in each of the following areas:

i. The cost of developing software. (2 marks)

If some of the requirements of a project can be met by models or software components that have been developed from an earlier project or from an outside supplier then time spent creating the code is saved although this saving will be partly offset by the cost of managing a library of reusable components or the cost of buying in the software.

ii. The quality of software. (2 marks)

If the developer can reuse a design or component that has been tested and proved to work in another application then there is a saving in the time spent to test and quality assure that component.

iii. The flexibility of software. (2 marks)

The ability to recombine information systems components or services in new ways to create new business processes or adapt existing ones is important to business stakeholders in a changing world.

b) Explain how the developers of an object oriented software system might make use of the following to encourage software reuse:

i. Object "wrappers". (3 marks)

The idea is to wrap legacy systems to make them appear to be object oriented – so a wrapper offers services that trigger functionality in the legacy application – sometimes this involves simulating the key strokes that would have been made by the user of the legacy application.

ii. Class Libraries. (3 marks)

A class library is a collection of prewritten classes or coded templates, any of which can be specified and used by a programmer when developing an application program through the mechanism of inheritance.

iii. Commercially available componentware. (4 marks)

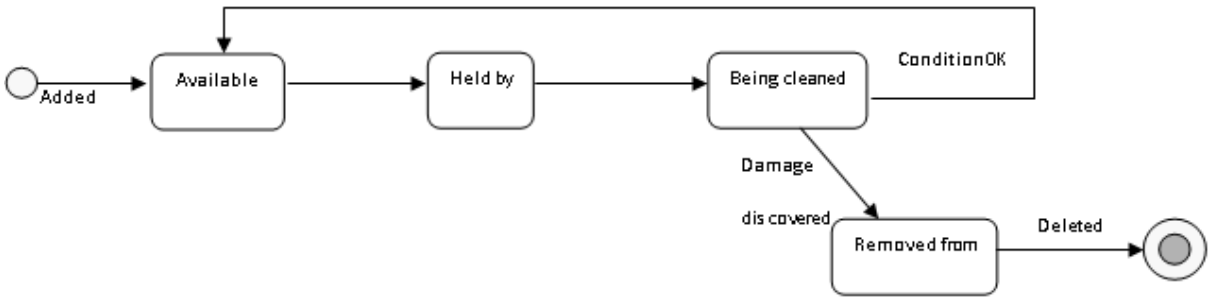
OO techniques allow us to develop components that offer services and can be incorporated in new software systems. Suppliers provide catalogues of components that provide the developer with the possibility of building a wide range of different functions into their applications without having to reinvent the wheel. Examples from the course text include: serial communications, computer-aided design, project management using Gantt Charts and spreadsheets.

iv. Web Services. (4 marks)

The use of web services is an increasingly important way of achieving software reuse. These allow one application to provide a service to other applications over the web, involving messages that are transmitted using a protocol such as SOAP. Many web services provide a machine-readable definition of the services that they provide.

Question B10 - Addresses L03.

a) Draw a statechart showing the various states that the "house" class might transition throughout its lifetime. You should identify at least **FOUR** states. Explain any assumptions you make in constructing this diagram. (10 marks).



The examiner will have to use some discretion in allocating marks for this answer – typically 2 marks for each correct state with the remaining marks being used for the transitions.

b) Explain why **ALL** the guard conditions from a state should be mutually exclusive. (4 marks).

All the guard conditions from a state should be mutually exclusive so that for each set of circumstances there is only one valid transition from a state [2]. If they are not mutually exclusive more than one transition may be valid and the behaviour of the state machine is indeterminate.[2]

c) Describe the **KEY** ways in which a statechart should be cross-checked for consistency with other UML diagrams. (6 marks).

A state machine should be cross-checked with relevant class diagrams to ensure that the class specifications are consistent, with interaction sequence diagrams, communication diagrams interaction overview diagrams and timing diagrams that involve this class (and hence its state machine), with any nested state machines and also with any linked activity diagrams (e.g. parent activity diagram).[3]

The consistency checks should include class names, location of operations and signature of operations. The state changes must be consistent with those that are indicated by the interaction diagrams (in particular sequence and timing diagrams).[3]

Question B11 - Addresses L04.

a) Give an example of a persistent and of a transient object that might be represented in a class diagram. Explain the **KEY** difference between the two [5 marks].

Each response should be worth 3 marks to a maximum of five marks.
Transient – any object that is used as a temporary value, for example created for a calculation, or a string object that is created temporarily, for example for an error message, or something as large as the calculator object that is used as an example in the course text.
Persistent – any object that is stored in an information system, e.g. customer, order, product, line, line fault, etc. The key characteristic of these objects is that their state is typically stored in a database table.

b) Explain **THREE** ways in which the classes in an inheritance hierarchy could be mapped to tables in a relational database. (6 marks).

1. Only implement the superclass as a table. Attributes of subclasses become attributes of this class and nulls are used to fill these values when they are not used. [2]
2. Only implement the subclasses as tables. Hold the attributes of the superclass in each table. [2]
3. Implement each class as a table. To retrieve the full set of attributes, data from the superclass table and one of the subclass tables must be retrieved. [2]

c) Explain **TWO KEY** differences between a relational DBMS and an object DBMS. (5 marks).

Each response worth 3 marks to a maximum of 5 marks.

Two possible key differences:

1. *For objects to be stored in a relational database they must be converted to flat two-dimensional rows of data. Complex objects can be stored as they are in an object database.*

2. *Object databases provide persistence mechanisms that are effectively transparent. For example, using ObjectStore, persistent objects only need to be created and amended within a transaction for them to be stored automatically. To store objects in a relational database, special calls must be made to software to handle the storage.*

d) What is meant by the acronyms OML and ODL in relation to object databases? (4 marks).

Object Manipulation Language [2] and Object Definition Language[2].A few words about what these mean should be provided for each.

Question B12 – Addresses L02

a) What is the **KEY** difference between "white box" and "black box" tests? (4 marks).

Black box testing checks the performance of a program or program module by starting with a particular input and generating an output. By knowing what the output should be from that particular input (based upon program specifications), the program can be assessed[2]. White box testing evaluates the actual code in the program or program module. Review of the source code can reveal

problems or incorrect assumptions that need to be corrected, particularly when the logic is complex.[2]

b) Identify **THREE** unit tests that could be applied during the design of an interface to an ATM (Automated Teller Machine). Describe how **EACH** of these tests should be documented. (6 marks).

One mark for each Unit Test and three marks for the generic description of the documentation requirements.

There are many tests that could be presented: Verify card insertion, verify PIN no. entry, verify operation with wrong PIN no.; verify Account type selection and so on.

The Unit Test documentation for these should include the following fields:
Test case ID; Test priority (Low/Medium/High); Test Designed By: Name of tester; Test Designed Date: Pre-conditions: Any prerequisite that must be fulfilled before execution of this test case. A list of all pre-conditions in order to successfully execute this test case; Dependencies: Mention any dependencies on other test cases or test requirement. Test Steps: A List all test execution steps in detail. Test Data: Use of test data as an input for this test case. Expected Result: What should be the system output after test execution?

c) Identify and describe **FIVE** approaches to system testing. (10 marks).

The purpose of requirements testing is to determine whether the original business requirements have been met [2]. The purpose of usability testing is to determine how convenient the system is to use from a user standpoint[2]. The purpose of security testing is to test disaster recovery processes and unauthorized access processes[2]. The purpose of performance testing is to examine the system's ability to perform under high loads[2]. The purpose of documentation testing is to determine the accuracy of the current documentation[2].

Question B13 - Addresses L01

a) Explain what is meant by the terms 'quality problems' and 'productivity problems' associated with systems development. (4 marks).

Quality problems affect the nature of what is achieved [2]. Productivity problems determine when, and whether, anything at all will be achieved[2].

b) Identify **FOUR** quality problems that may occur in systems development. Comment on the ways in which we can minimize the risk of each problem occurring. (8 marks)

Flynn gives these as the main reasons. Any four of these will do for 2 marks each:

- *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.*

The examinee should explain that these are the types of problem addressed by system development methodologies. The use of models that clients can understand and an emphasis on iterative incremental development and regular meetings with stakeholders are all intended to address these problems.

c) An online video streaming service allows registered customers to watch films on a pay-per-view basis. It has been suggested that the system supporting this activity should be linked to the organisation's management information system. Identify and briefly describe **FOUR** ethical issues that might be involved in integrating the two systems. (8 marks)

One obvious issue here is the privacy of the customer[2]—in what ways will their purchases and shopping behaviour be stored, analysed and reported?[1] Who will use this information, and how? [1] On-line shopping in itself raises other questions, some of which are at least partly ethical in scope. For example: what is the nature of the community in which shoppers will engage, and how does this affect social relationships in the wider world?[1] What is the effect on levels of employment within the organization? [1]How is the working environment changed? [1] What are the effects on career possibilities for employees? [1]

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