

UNIVERSITY OF LONDON

291 0226 ZA

B.Sc. Examination 2008

COMPUTING AND INFORMATION SYSTEMS

**CIS226 Software Engineering, Algorithm Design and Analysis
(Western)**

Duration: 3 hours

Date and time Wednesday 14 May 2008 : 10.00 – 1.00 pm

Full marks will be awarded for complete answers to a total of FOUR questions.

You must answer TWO questions from Section A and TWO questions from Section B.

Each question carries 25 marks. The marks for each part of a question are indicated at the end of the part in [.] brackets.

There are 100 marks available on this paper.

No calculators should be used.

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

Section A

Question 1

- (a) Give a short account of why requirements engineering is a difficult process. [4 marks]
- (b) What is the relationship between use cases A and B if:
- (i) use case A *includes* use case B [2 marks]
 - (ii) use case E *extends* use case B [2 marks]
- (c) Answer the following questions regarding classes, interfaces and abstract classes.
- (i) How are *interfaces* different from *classes*? [2 marks]
 - (ii) How are *abstract classes* different from *interfaces*? [2 marks]
 - (iii) What is the relationship between an interface and a class that *realises* or *implements* it. [2 marks]
 - (iv) Define the term *instantiation*. Can a class be *instantiated*? Can an interface be *instantiated*? Can an abstract class be *instantiated*? [4 marks]
- (d) Dependency is a relationship between class definitions.
- (i) With reference to source code modifications, explain what it means that *class A depends on class B*. [2 marks]
 - (ii) How is dependency between two classes, say A and B, represented diagrammatically in UML? [2 marks]
 - (iii) With reference to messages, attributes and operations (methods), state in which circumstances *class A depends on class B*. [3 marks]

Question 2

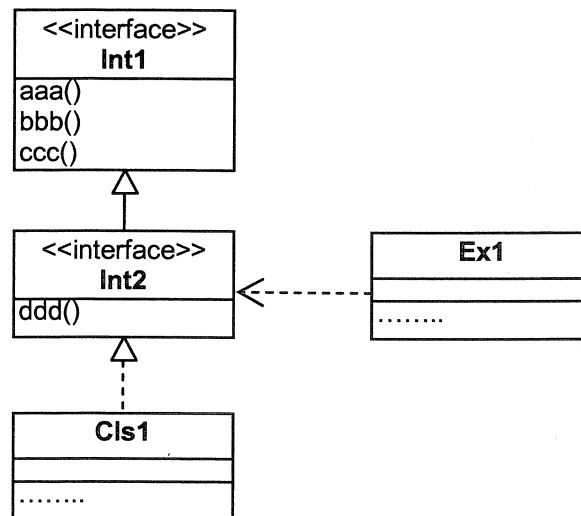
- (a) Draw a use case diagram for the following specification for an internet banking system. *Users should be able to make payments and view statements on line. In each case, i.e. before viewing statements or making a payment, a user will have to undertake an identification procedure. If a payment would cause the account to become overdrawn, the payment is rejected.*

[5 marks]

- (b) Represent the following specification for a banking application as a UML class diagram. *There are two types of accounts, Current and Savings. Each customer can have at most one account of each kind. Current accounts can be shared, but by not more than two customers. Savings accounts cannot be shared.*

[6 marks]

- (c) Consider the following class diagram.



- (i) What should *Cls1* necessarily include as code components when it is implemented in an OO programming language? [2 marks]

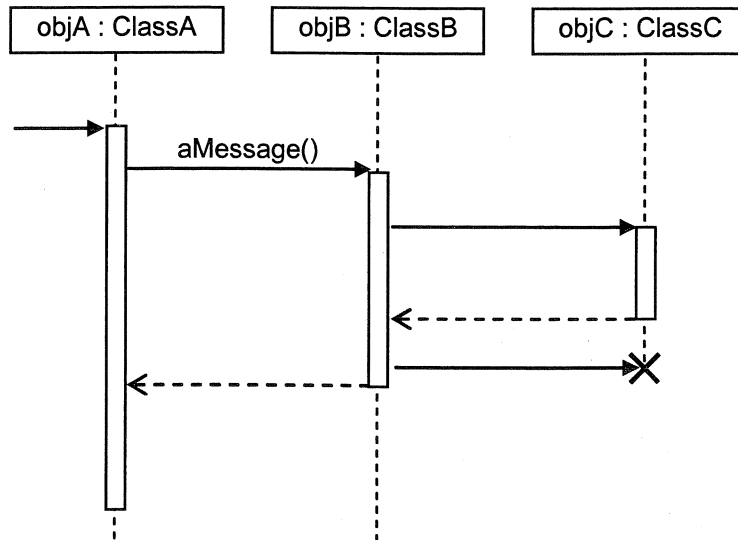
- (ii) Which parts of code (i.e. corresponding to which diagram components) should or might be required to be modified, if a modification is made to the code representing interface *Int1*, and which are safe to remain unchanged? Explain your answer. [4 marks]

- (d) Represent the following specification for a computer racing game as a UML class diagram. *A car is modelled with two essential components: an engine system and a control system. One car has one engine system and one control system. A car is no longer a car if any of these two components is missing. A car may also have (but this is not necessary) driver settings, which affect its behaviour – i.e. the way the engine system responds to the control system. One car can have more than one driver setting.*

[8 marks]

Question 3

(a) Consider the following sequence diagram.



Answer the following questions.

- (i) What do the rectangles at the top of the diagram represent? [1 mark]
- (ii) What do the vertical dotted lines represent? [1 mark]
- (iii) What do the vertical rectangles represent? [1 mark]
- (iv) What does it mean that an object has a *live activation*? [1 mark]
- (v) Is time represented in this diagram? How? [2 marks]
- (vi) What do the cross (X) and the arrow that meets it represent? [2 marks]
- (vii) Which class should implement the method (operation) *aMessage()*? [1 mark]

(b) Consider the following specification.

A simple drawing application consists of a tool bar and a canvas. The canvas can contain basic shapes, such as circles and rectangles. Among other tools, the toolbar includes a shape selection, a colour selection, and a paint tool. A shape is filled with a colour by: selecting the shape, selecting a colour and then clicking on the paint tool.

Consider that the canvas contains only two shapes, one circle and one rectangle, and the rectangle has already been selected (with the selection tool). Consider also that a colour has also been selected. Draw a sequence diagram which describes the process of colouring the rectangle. **Hint.** There are four objects in this scenario: a paint tool, a canvas, a circle and a rectangle. The paint tool is live activated with a *click()* message from the user. The first message sent by the paint tool is to the canvas, to enquire which shape is selected. The canvas asks first the circle and then the rectangle. [16 marks]

Section B

Question 4

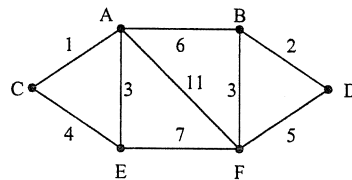
- (a) Explain what is an *optimisation problem*. [2]
- (b) Describe the ideas of the *Huffman encoding* algorithm. [2]
- (c) Discuss two different instances of the Huffman encoding problem including one *special case*. [4]
Hint: A special case can be a situation where there is no solution or that requires an unusual solution. Add assumption if necessary.
- (d) Design and outline an algorithm for the Huffman encoding problem. Explain the data structure used, and the input and output expected. [12]
- (e) Analyse the time complexity of your algorithm for the worst case. [5]

Question 5

- (a) Consider the task of maintaining a record list of income-tax returns. It requires keeping track of the dependents of each of potentially millions of people.
 - (i) Propose a suitable data structure for the task. Justify your choice and add assumptions if necessary. [3]
 - (ii) Discuss the suitability of using two standard methods *size* and *isEmpty* in terms of efficiency. [4]
 - (iii) Suppose that we need to keep track of the total number of the customers. How would you implement the two methods? [3]
- (b) The terms *abstract data structure* and *abstract data type* are often used interchangeably. Explain, with an example, when *abstract data structure* is more appropriate and when *abstract data type* is more appropriate. [5]
- (c) Explain, with an example, how to distinguish a *binary heap* and a *binary tree*. Draw both types of tree structures for (5, 7, 10, 6, 1, 14, 11, 2) in your example. [5]
- (d) Explain, with an example, what a *compressed trie* is. Demonstrate how the following words can be stored in the data structure. [5]
(visit, very, roll, splay, root, rotation, vector, suger, vertex).

Question 6

- (a) Explain, with an example, why every node of a binary tree should be an *object* type containing at least two different types of fields in implementations. [5]
- (b) Explain the principle(s) of dynamic programming. Write an algorithm `dynamicFibonacciItem(n)` in pseudocode or Java to compute the n th Fibonacci number, applying the dynamic programming principles. Describe the advantages of the dynamic-programming approach over the recursive approach. [10]
- (c) Draw a minimum weight spanning tree of the simple graph below and demonstrate a *preorder traversal* of the tree by writing down the vertex labels in the order of being visited, starting from vertex E. [5]



- (d) Write a method `equal(L1, L2)` for comparing the contents of two linked lists. It should return `true` if both lists contain equal values. Explain how your algorithm works by giving a small example. [5]