THE OPEN UNIVERSITY OF SRI LANKA
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
ECX4262 – OBJECT ORIENTED DESIGN AND PROGRAMMING

FINAL EXAMINATION - 2009/ 10   CLOSED BOOK

Date: March 09, 2010   Time: 1330 – 1630 hrs

Answer Q1 and any other THREE questions from Q2 – Q6.

Q1   (Compulsory question)

A department store employs a number of sales assistants. Sales assistants are normally stationed at tills. The first thing sales assistants do when they arrive at a till is to log on to the sales system to which all tills is connected. Logging on identifies them to the system. At the end of the day, or whenever they are moved to another department, the sales assistants log off the till. Only recognized users may log on to a till. The System Manager registers users with the system. The System Manager may also remove users from the system. When a customer wishes to purchase an item, they approach sales assistant who receives a payment from the customer (which may be in the form of cash, credit card or cheque) enters it into the till and records the item that has been purchased. Occasionally, customers are dissatisfied with the purchases they have made and return them. They take them to a sales assistant who refunds their money and records the return of the item via the till. The advantage of this system to the department store is that it enables managers to get regular sales reports showing how well given items are selling.

(a) Who are the main actors in this system? [04]

(b) Draw a Use case diagram for the system. [12]

(c) Draw an Activity diagram for the system. You should show the swimlanes. [12]

(d) Draw a Class diagram showing the main classes and their associations with the relation names. You should include all cardinalities. You should also include any obvious opportunities for inheritance. [12]

Q2

(a) [7]
   i)   Explain the terms *object*
   ii) Explain the terms *class*
   iii) What is the relationship between a *class* and an *object*? Illustrate your answer with Java/ C++ code.

(b) [7]
   i)   Describe what is meant by the term *instance method*;
   ii) Describe what is meant by the term *class method*;
   iii) Explain where it is appropriate to use an *instance method* and where it is appropriate to use a *class method*.

(c) Write Java/ C++ code to implements the class 'Coin' as follows. [6]

The class Coin represents a monetary coin. Each coin has a monetary value (recorded as a whole number) associated with it.

The class has two instance methods setValue and getValue which respectively set and return the value associated with the coin.

The class has a class method that returns the current number of Coin objects.

Q3

(a) Distinguish between the terms subclass and superclass. [2]

(b) Why is a class hierarchy important when modelling object oriented systems? [3]

(c) What do you understand by the term abstract class?
Why the classes change into abstract classes, explain with an example. [9]

(d) What do you understand by the term interface class? [2]

(e) Why object oriented software should be developed in terms of interfaces. [4]

Q4

(a) Explain each of the following concepts. [8]

   i)   constructor   ii)   method overloading

   iii)   method overriding   iv)   polymorphism

(b) In an object oriented programming language with which you are familiar, provide sample code that demonstrates the use of EACH of the above concepts. [12]

Q5   Draw <u>sequence diagrams</u> for the following:
An automated teller machine (ATM) is a computerized device that provides a bank's customers a secure method of performing financial transactions in a public space without the need for a bank clerk. Having inserted a valid card and password, a user should be able to:

(a) Withdraw money from the ATM. [10]

(b) Get a display the current balance. [5]

(c) Change the password by entering the new details twice. [5]

Q6

(a) What is meant by the term *pattern* in the context of OO development? [2]

(b) What are the three types of Design patterns? [3]

(c) Give three methods for each design pattern. [3]

(d) Select ONE method from EACH design pattern and give a detailed description of each, stating the problem they address and the basis of the solution they offer. [12]