University of Swaziland

Department Of Computer Science

Main Examination

DECEMBER 2014

Title of paper : ***C under Unix***

Course number : ***CS344***

Time Allowed : ***Three(3) hours***

Instructions :

- ***Each question is worth 25 marks***
- ***Answer question 1***
- ***Answer any three(3) questions from questions 2 to 6***

This paper may not be opened until permission has been granted by the invigilator

Question 1-Compulsory-25 marks

(i) With the aid of C++ or C# or Java code examples, explain the meaning of the following object oriented concepts.

(a) Object [2]

(b) Class [2]

(c) Inheritance [2]

(ii) With the aid of examples, distinguish between

(a) Overloading and over-riding [3]

(b) Single inheritance and multiple inheritance [3]

(c) Abstract class and concrete class [3]

(iii) Consider the program fragment given below.

```
Class Set<T> {
                void  insert (T newElement);
                void  remove(T anElement);
};
Set <Employee>  employeeSet;
```

(a) What is the role/meaning of <T> in the declaration above. [3]

(b) Explain the meaning of the last statement in the above code. [3]

(c) Show how the above code segment would appear in the three different languages considered in the course. [C++, C#, Java] [4]

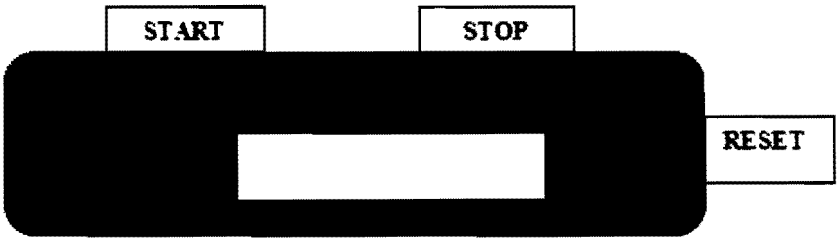
Question 2-25 marks

Draw a UML Class Diagram representing the following elements from the problem domain for a football league.

A football league is made up of at least twelve football teams. Each football team is composed of six to twelve players, and one player captains the team. A team has a name and a record. Players have a number and a position. Football teams play games against each other. Each game has a score and a location. Teams are sometimes lead by a coach. A coach has a level of accreditation and a number of years of experience, and can coach only one team. Coaches and players are people, and people have names and addresses. Draw a class diagram for this information, and be sure to label all associations with appropriate multiplicities.

Question 3 – 25 marks

Consider the following simple integer counting device.



The device has 3 buttons: **Start**, **Stop** and **Reset**. The count value is displayed on the screen labeled count value. The **Start** button starts the counting from zero or from the last count before the counting was stopped. The Stop button stops the counting and the display reflects the last value. The reset button resets the count value to 0.

Consider a program that could be used to simulate this simple counting device. And,

(i) Using a UML diagram, draw an Object-Oriented Analysis (OOA) class diagram that shows the human interaction component (HIC) [User Interface component] of the counting device. [6]

(ii) Using C++ or C# or Java notation/terminology, re-draw the diagram obtained in (i) above to show how it can be implemented as a graphical user interface (GUI) in the selected language. [3]

(iii) How many classes would be needed to implement the problem domain component (PDC) for the counting device? [2]

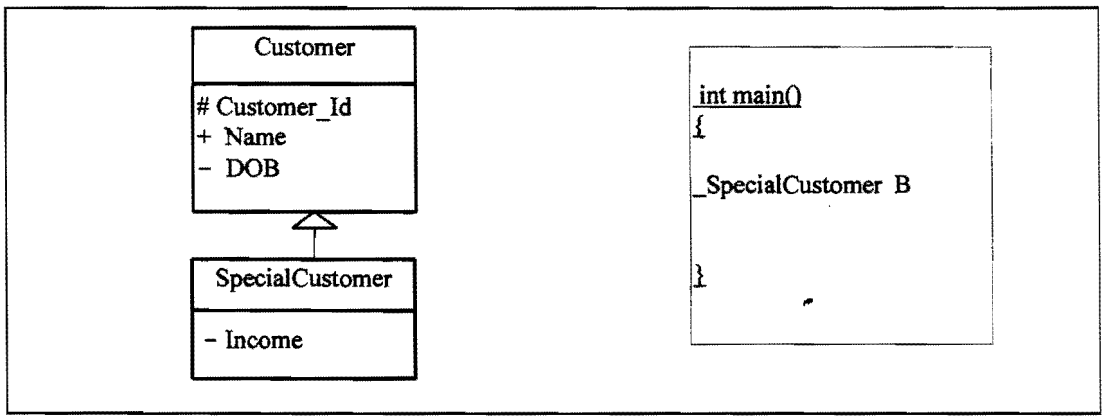
(iv) Draw a simple class diagram containing the details of the classes obtained in (iii) above. [4]

(v) Using C++ or C# or Java notation, describe the classes obtained in (iv) above. All required constructor, data members and member functions must be declared. [6]

(vi) Using pseudocode, describe what objects and event handlers need to be declared in the main program that implements the simple counting device. [4]

Question 4 – 25 marks

(i) Consider a system that has a class called **Customer**, and a sub-class called **SpecialCustomer** as shown in the following UML model. A simple main() program is also shown on the right.



Determine if each of the following statement are **true** or **false**. In each case, briefly explain your answer

(a) An attempt to print the value **B.Customer_Id** from the main program will produce a syntax error. [2]

(b) A member function **Display()** defined for the SpecialCustomer class can directly print out the Customer_Id but not the DOB. [2]

(c) A public member is visible anywhere in the program and may be accessed by any object within the program. [2]

(d) A private member is visible only to class that defines it, and its sub-classes, but not to any other class, even a "friend" class. [2]

(e) If private inheritance is used, an attempt to print B.name in the main function will not cause an error. [2]

(ii) Using C++ or C# or Java notation, write complete definitions for the Classes Customer, and SpecialCustomer. Each class definition must contain a Display() function that prints out the values of the attributes of any object of that class. [15]

Question 5

Using the C++ Standard Template Library (STL) or C#/Java Collection, write a program that implements an array, L [0..N] of linked lists. The program must read a sequence of integers from a text file. For each integer X read from the text file the program must insert X into linked list L[I] only if X is a multiple of I. Once the program has read all the integers from the text file it must display all the values in each linked list on standard output.

Question 6

Consider rational numbers (fractions) of the form: **a/b**, where **a** is the numerator and **b** is the denominator. The arithmetic operations addition, subtraction, multiplication and division can be performed on rational numbers. These operations can be performed as shown in the following examples.

a/b + c/d = (ad + bc) / bd
a/b - c/d = (ad - bc) / bd
a/b * c/d = ac/bd
(a/b) / (c/d) = ad/bc

(i) Using C++ notation, define class ***Rational*** for performing arithmetic operations with fractions. Use integer variables to represent the numerator and the denominator of a fraction. In the definition of the class provide prototypes for member functions to perform the arithmetic operations addition, subtraction, multiplication and division. Use operators instead of functions as much as possible. For example use operator + instead of function name **add**. The class should also specify a member function (or operator) for printing a rational number in the form **a/b,** where **a** is the numerator and **b** is the denominator. Your class should have two (2) constructor functions. The first constructor should have no arguments and simply initializes the numerator and denominator to be 0 and 1 respectively. The second constructor takes two (2) integer arguments and initializes the numerator and denominator to be the first and second arguments respectively. *10 marks*

(i) Using C++ notation, write the code (definition) of the member functions whose prototypes are specified in the definition of class ***Rational*** above. *15 marks*