MSc in Computer Science

MSc in Computer Science with Data Analytics

MSc in Computer Science with Cyber Security

MSc in Computer Science with Artificial Intelligence

DEPARTMENT OF COMPUTER SCIENCE

SOFTWARE ENGINEERING

Time restricted exam

Time allowed: 120 (one hundred and twenty) minutes plus an additional 30 (thirty) minutes for file uploads.

The exam requires you to upload some of your answers. Hand written answers are acceptable. Alternatively, you may also use a software tool of your choice. Your uploads must be in either a PDF, a JPG or a PNG format.

The exam paper will total up to 100 marks, and will cover the following sections:

Section A: Object Constraint Language

Section B: Use Case and Class Modelling

Section C: Interaction and State Modelling

Section D: Software design, testing and management

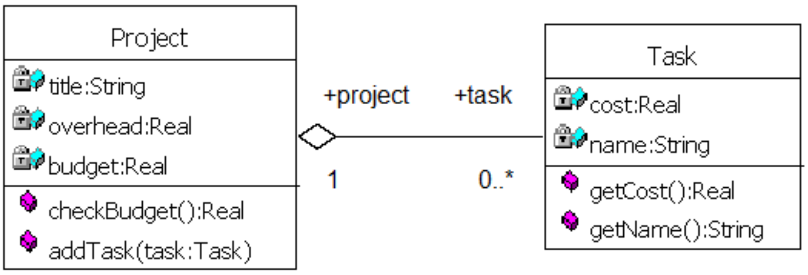
Answer all the questions.

Please read the following instructions before attempting the mock exam:

Instructions for the Software Engineering exam.pdf

Section A Object Constraint Language [30 marks]

A project contains a number of tasks. Each project has a unique title, a budget and an overhead cost. Each task has a name and an associated cost. The task cost and name can be accessed externally via the *getCost and getName* operation. The project budget can be checked via the *checkBudget* operation and a new task can be added via the *addTask* operation. The following diagram gives the skeleton of a UML class diagram describing the projects and tasks. In the class diagram, all attributes are *private* and all operations are *public*.



Write down the OCL expressions that represent the following requirements:

1. [8 marks] There are two pre-requisites for a project: its budget should be at least twice of its overhead cost and the length of the project title should be no more than 20 characters.

2. [10 marks] The operation *checkBudget* is used to check the budget for a project. It has one pre-requisite that there should be at least 3 tasks associated with that project. It has one post-requisite that the operation returns the difference between the budget and the project cost (project cost is the sum of the associated advert costs plus the project overhead cost).

3. [12 marks] The operation *addTask* is used to add a new task to a project. It has two pre-requisites that the task is not yet part of the project and after adding the task the project cost will not exceed the budget. It has one post-requisite that the task is successfully added to the project.

Section B Use case and class modelling [30 marks]

You are asked to develop a learning activity management system for a secondary school at York during the time of home schooling. The system is to be used by the teachers (identified by *staffID* alongside *user name*) and all the students of the school (identified by *studentID* alongside *user name*), and managed by a system administrator. The system should maintain a list of all subjects offered by the current academic year. A subject identifies itself by a name and the year group (e.g. Y7 Maths) and contains a number of learning activities.

Due to the usual resource constraint, one subject is taught by exactly one teacher, and a teacher is responsible for one to three subjects. Upon receiving the paper form from the head teacher on "*who teaches what*", the system administrator allocates teachers to subjects.

A subject teacher can create a number of learning activities for each of the subjects they are responsible. The relevant students can be assigned to a learning activity upon the creation of the learning activity, or later at the subject teacher's discretion. Once a learning activity has been assigned, an email should be sent via an external email system to notify students. Immediately after a learning activity is done, the subject teacher enters the number of actual participants to the system.

A student takes a number of subjects and attends all the learning activities for the subjects taken. They can view the details of learning activities assigned to them, e.g. location, date/time.

The system is required to manage private activities as well as subject learning activities discussed above. Private activities (e.g. play piano) are created, managed by individual teachers or students. As well as a date and time, each private activity also has a priority attribute with the value 'high', 'medium' or 'low', a reminder email can be sent by the external email system according user preferences 1 hour before the start of the activity (e.g. send reminder email for 'high' priority activity).

The system stores all the notification/reminder emails (e.g. receipts and message) sent to the students. The system administrator manages all the teachers and students.

4. [10 marks] Draw a use case diagram solely based on the information given above. You do not need to show the use case that enables a user to log onto the system.

5. [20 marks] Draw a conceptual class diagram solely based on the information given above. Your class diagram should show entity classes, attributes, associations and multiplicities. You don't need to show boundary, control classes, operations and data types on your class diagram.

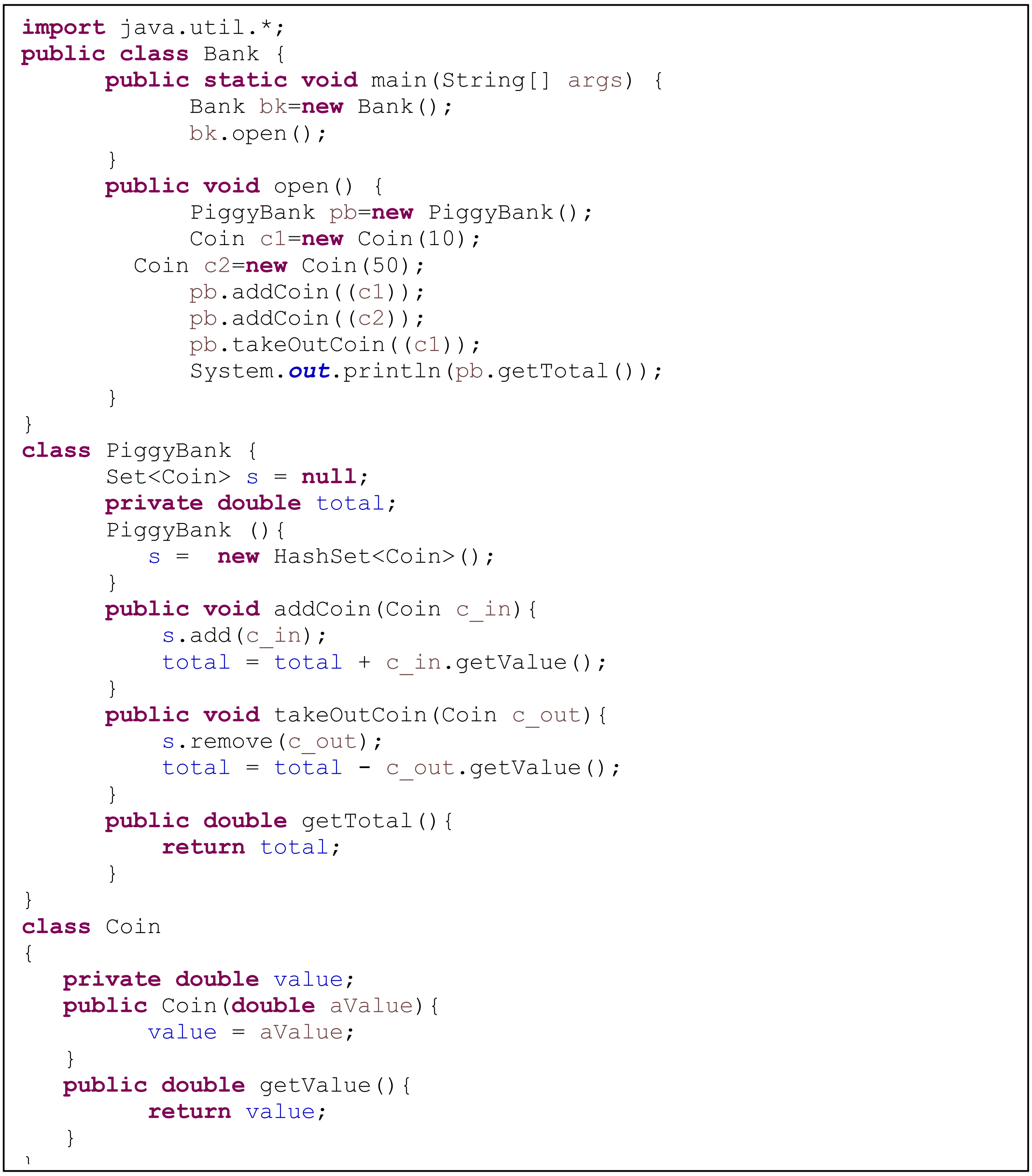
Section C Interaction and state modelling [20 marks]

6. [8 marks] A safe has a combination lock that can be in one of three positions, labeled 1, 2, and 3. The dial can be turned left or right (L or R). Thus there are six possible dial movements, namely 1L, 1R, 2L, 2R, 3L, and 3R. The combination to the safe is 3L, 1L, 2R; any other dial movement will cause the alarm to go off. The state transition table is given below.

Current state Resulting state Dial movement	Safe locked	S1	S2
1L	Sound alarm	S2	Sound alarm
1R	Sound alarm	Sound alarm	Sound alarm
2L	Sound alarm	Sound alarm	Sound alarm
2R	Sound alarm	Sound alarm	Safe unlocked
3L	S1	Sound alarm	Sound alarm
3R	Sound alarm	Sound alarm	Sound alarm

With *safe locked* as the initial state, draw a state diagram representing the states of the safe and the transitions between them.

7. [12 marks] Read the Java code below and produce a sequence diagram for the *open()* operation in the *Bank* class. You are not required to show returning messages on the diagram.



```java
import java.util.*;
public class Bank {
      public static void main(String[] args) {
            Bank bk=new Bank();
            bk.open();
      }
      public void open() {
            PiggyBank pb=new PiggyBank();
            Coin c1=new Coin(10);
        Coin c2=new Coin(50);
            pb.addCoin((c1));
            pb.addCoin((c2));
            pb.takeOutCoin((c1));
            System.out.println(pb.getTotal());
      }
}
class PiggyBank {
      Set<Coin> s = null;
      private double total;
      PiggyBank (){
         s =  new HashSet<Coin>();
      }
      public void addCoin(Coin c_in){
          s.add(c_in);
          total = total + c_in.getValue();
      }
      public void takeOutCoin(Coin c_out){
          s.remove(c_out);
          total = total - c_out.getValue();
      }
      public double getTotal(){
          return total;
      }
}
class Coin
{
   private double value;
   public Coin(double aValue){
         value = aValue;
   }
   public double getValue(){
         return value;
   }
}
```

Section D Software design, testing and management [20 marks]

8. [4 marks] Identify the key strength and weakness of the closed layered architecture.

9. [4 marks] with no more than 100 words, identify and justify the two most important personal characteristics that you believe a successful software quality assurance manager must have.

10. [8 marks] This is a two-part question - Part 1:
Draw a control flow graph to represent the *guess()* method in the *HiLo* class. You are required to use a rectangle to represent a sequence of uninterrupted program statements, a diamond to represent a decision, and a circle to represent a junction.

```
import java.util.*;
public class HiLo {
public static void guess(){
   int target=10;int guess;
   do{
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter you guess "
        guess = sc.nextInt();
        if (guess>target)
            System.out.println("too high");
        else if (guess<target)
            System.out.println("too low");
      }while (guess!=target);
 }
 public static void main (String args[]){
     guess();
     System.out.println("Hit, well done!");
 }
}
```

11. [4 marks] You are then required to write a minimal set of tests that can achieve the branch coverage for the *guess* method. Please write the set of tests in the form of
T1 = {guess 1},
T2 = {guess 2},
…
where guess 1, guess 2…are the guess number entered by the user of the program.

Model Answers

Section A Object constraint language [30 marks]

[Flexible for different expressions achieving the same effects]
Partial marks can be allocated as appropriate

Question 1 Solution [8 marks]:

```
context Project                                         [2 marks]
   inv: budget>=2*overhead                              [3 marks]
   inv: title.size()<=20                                [3 marks]
```

Question 2 Solution [10 marks]:

```
context Project::checkBudget():Real                     [2 marks]
   pre:task->size()>=3                                  [3 marks]
   post: result=budget-overhead-task.getCost()->sum()   [5 marks]
```

Question 3 Solution [12 marks]:

```
context Project::addTask(t: Task)                       [2 marks]
   pre : task->excludes (t) --or not task->includes (t) [3 marks]
   pre : budget >=task.getCost->sum()+ t.getCost()      [4 marks]
   post: task->includes (t) --or task=task@pre->including (t)   [3 marks]
```

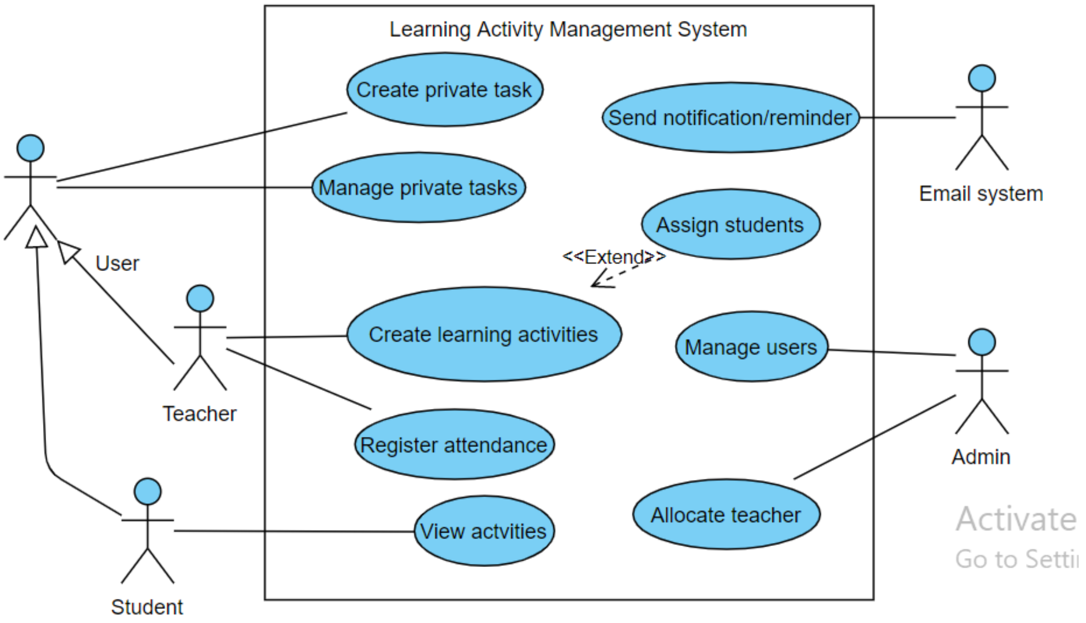
Section B Use case and class modelling [30 marks]

Question 4 Solution [10 marks]

4 marks for correct identification of the actors (Teacher, Student, Admin and Email system), 1 each.

1 marks for the *user* actor and the correct representation of the actor generalisation relation.

5 marks for the coverage of all major functions and the extend relation, half each.



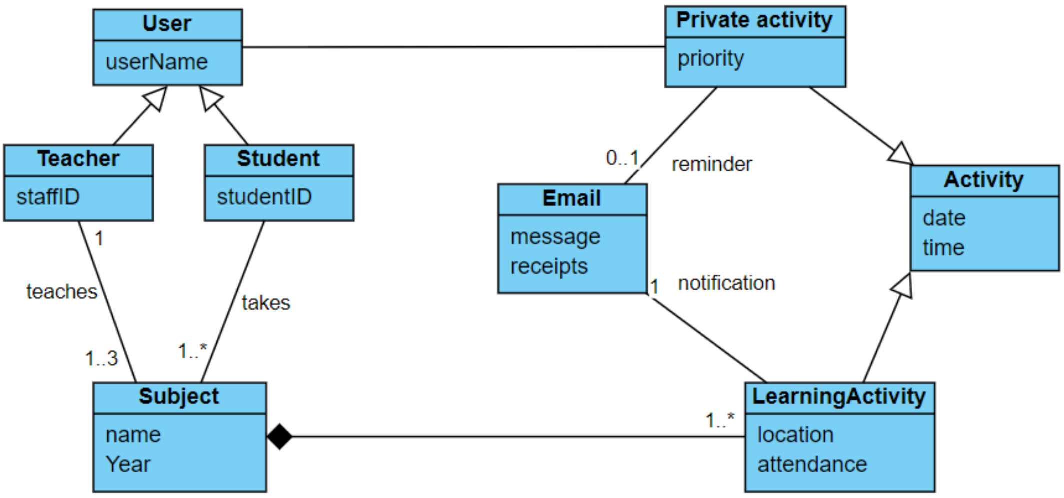
Question 5 Solution [20 marks]

8 marks for the correct identification of classes: 1 mark each.

4 marks for the correct identification of attributes for each class: 0.5 mark each.

5 marks for the correct use of associations and naming: composition 1, generalisation 2 and the rest 2.

3 marks for multiplicities, half each.

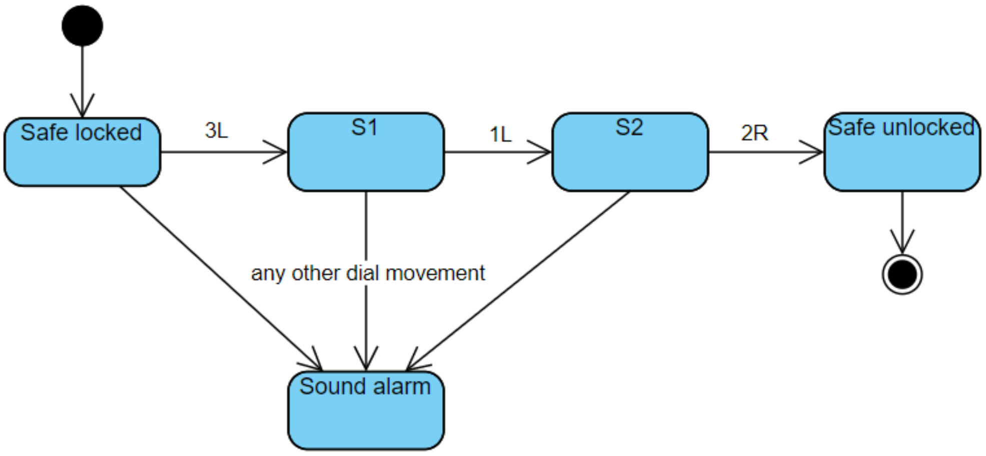


Section C Interaction and state modelling [20 marks]

Question 6 Solution [8 marks]:

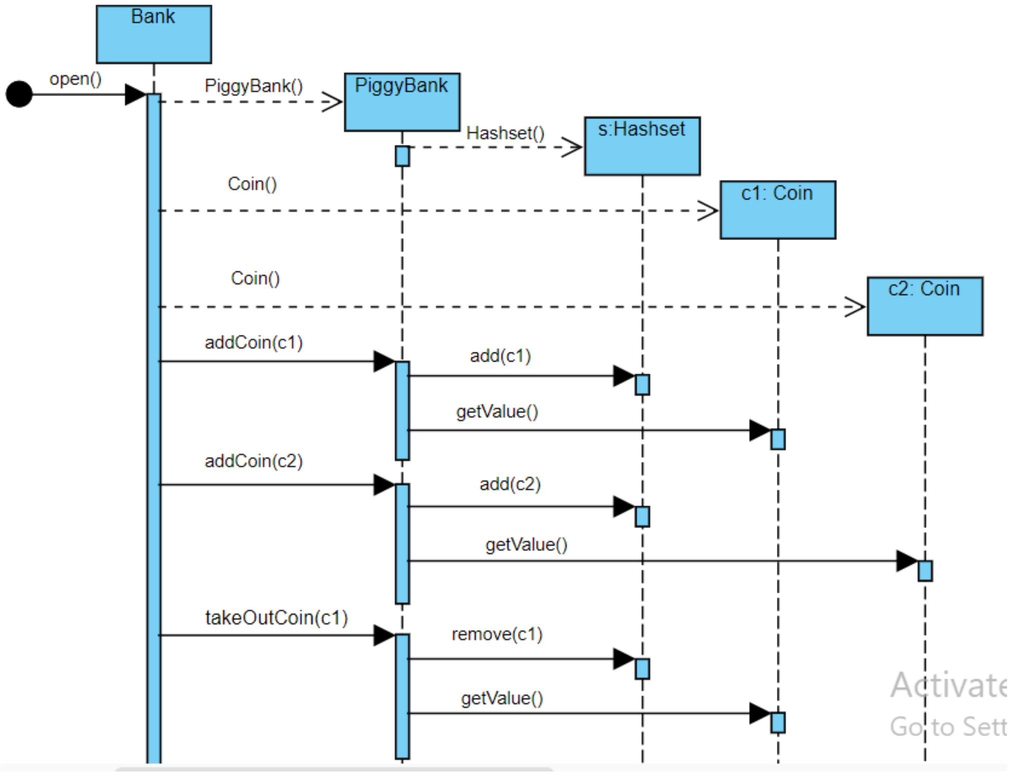
4 marks for the correct labelling of states, as a guideline, reduce 1 mark for each incorrect or missing major state.

4 marks for the correct transitions with correct labelling of events, as a guideline, reduce 1 mark for each incorrect transition.



Question 7 Solution [12 marks]:

Message could optionally show parameters.
5 marks for the correct labelling of participating objects, 1 mark each.
7 marks for the correct labelling of messages, e.g. correct calling and receiving object, correct timing for activation box, half mark each.



Section D Software design, testing and management [20 marks]

Question 8 Solution: [4 marks]

The key strength and weakness of the closed architecture.

- *Strength: low coupling*
- *Weakness: possible performance bottleneck issue*

2 marks each

Question 9 Solution [4 marks]:

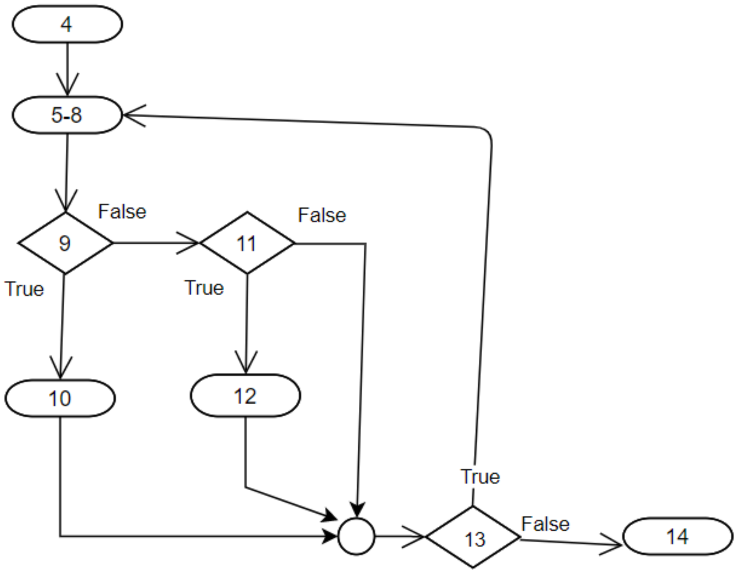
Identify and justify two personal characteristics for SQA manager

Any two of the following or similar will do:

- *Leadership*
- *Communication without offence*
- *Attention to details*
- *Technically competent*
- *Objective*

2 marks each with a short explanation that makes sense.

Question 10 Solution [8 marks]:



The key is the logic flow and branches, 2 marks for the correct representation each of the following:

- 11 to joint,
- 13 to 5-8,
- 9-10
- 11- 12

Question 11 Solution [4 marks]:

Need two tests:
one: >10
two:<10
For example, T1={11}, T2={9}

Award either 0 or full marks