OOSD Class Test 1 --- 29-08-2023

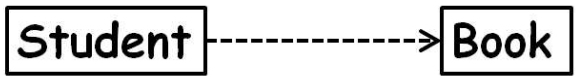
Time 55 mins. Marks 36

Name:

Roll Number:

---

1. In the context of developing a Library Automation System consider the following UML class diagram. Identify the relationship that exists between **Student** and **Book** classes in Write Java code for the **Professor** class that would lead to the given relation. **[1+2]**



Ans: **Relationship: Dependency ... Award 1 Mark**

1

```
class Student {
Book c;
public   void  borrow()  {  Book c = new Book();  ……..}
……..
}
```

2

```
class Student {
Book c;
public void  borrow(Book c)  {  ……..}
……..
}
```

Other reasonable methods such as reserve() etc should be accepted.

3

```
class Student {
Book c;
public void  borrow(Book c)  {  ……..}
……..
}
```

If Any of 1,2,or 3 answered: Award 2 Marks
1 Mark penalty for every mistake

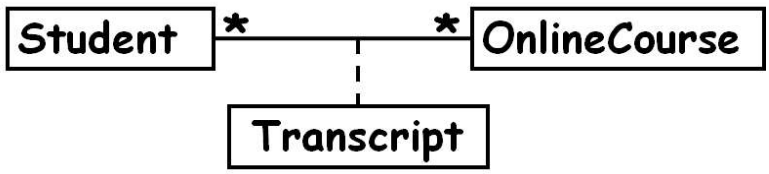
2. Consider the following Java code. Draw a UML 2.0 class diagram corresponding to the given code. **[4]**

```
class Student {
   private  ArrayList <Transcript> transcripts = new ArrayList <Transcript>();
}

class Transcript {
    Student student;
    Course course;
    Date subscriptionDate;
}

class OnlineCourse {
    private  ArrayList <Transcript> transcripts = new ArrayList <Transcript>();
```

}

Ans:

Deduct 1 mark for every mistake



OR

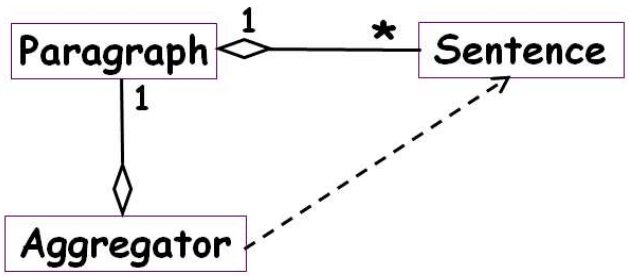
If the student has answered only the following diagram, maximum that can be awarded is 3. Also, 1 Mark penalty for every mistake

3. Consider the following Java code. Identify the class relationships evident in the cod and document them in UML 2.0 syntax. **[5]**

```
Public class Sentence{
  String text;
   Sentence(String ){text=s;}
}

Public class Paragraph{
   private ArrayList <Sentence>
        sentList=new
              ArrayList<Sentence>();
   public addSentence(Sentence s){
      sentList.add(s);
  }
}
```

```
Class Aggregator{
  public static void main() {
      BufferedReader br=new
      BufferedReader(new
      InputStreamReader(System.in));
      Paragraph p= new Paragraph();
      do{
          p.addSentence(new
          Sentence(br.readLine());
       }
   }
}
```

Ans:

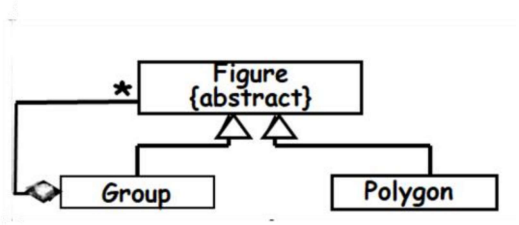


Apply 1 mark penalty for each Mistake (class, relation, or multiplicity)

4. Draw a UML class diagram corresponding to the following Java code. **[5]**

```
Abstract public class Figure {
       abstract public void draw();
       protected Pos position;
}

public class Group extends Figure {
      private Vector <Figure> figures = new Vector <Figure> ();
      public void draw () {
     }
}

public class Polygon extends Figure {
        public void draw () {        // draw polygon code
     }
}
```

Ans:

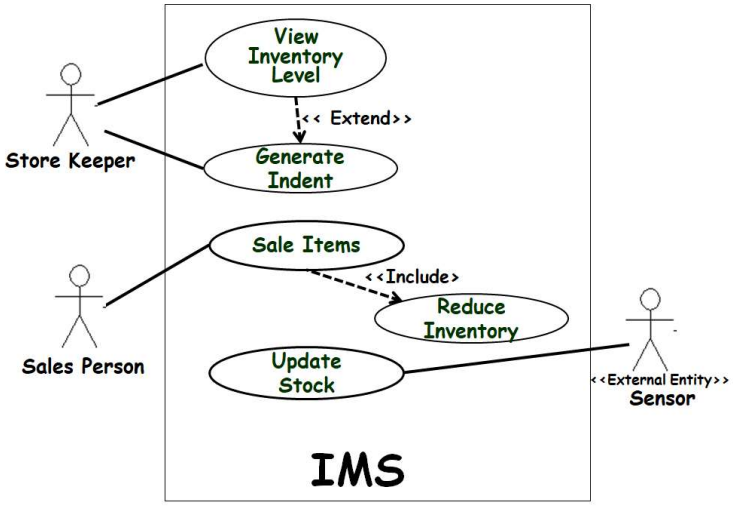


Apply 1 mark penalty for each Mistake (class, relation, or multiplicity)

5. Develop the use case diagram for the following web-based Inventory Management System (IMS). You need not provide a text description for your use case diagram. **[5]**

Using IMS, the store keeper should be able to view the inventory levels of specific productsk. IMS should allow the store keeper to generate an indent for the goods whose stock he observes to be below a threshold. While generating the indent, it should be possible to view the stock report if desired. The sales person should be able to invoke a sale transaction when an item is sold. When a sales transaction on completes, a reduce inventory function should be automatically invoked. When supply for any goods arrives, the goods are scanned by a hand held sensor one by one. The scanner machine automatically updates the stock in IMS through a Bluetooth connection.

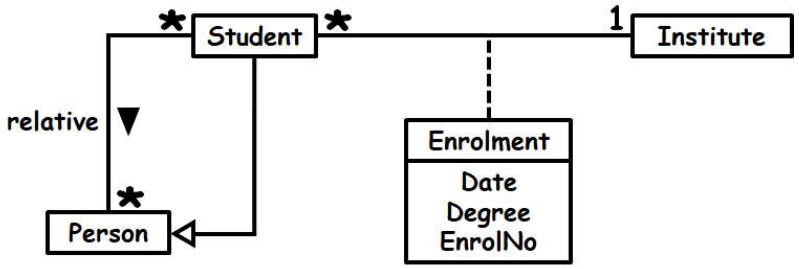
Model answer:



Apply 1 mark penalty for each Mistake

6. Develop a class diagram to model the following: "A student is a person who is enrolled with an Institute for a degree. Each student enrolment has a date of enrolment, degree enrolled, and an enrolment number. A student can have many persons as relatives and a person can have many students as relative. An Institute enrols many students" **[6]**

Ans:

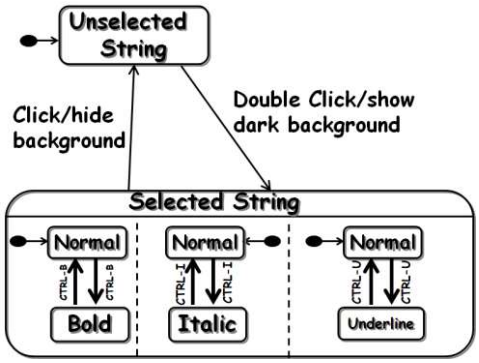


Apply 1 mark penalty for each Mistake

7. Design a UML state machine diagram to represent the following behaviour of the fonts of a character string pertaining to its appearance. **[8]**

A character string is by default not bold, not italicized, and not underlined. One can select a string by double-clicking the left mouse button on it, it appears with a dark background once selected. It is possible to change the appearance of a selected string as follows. One can independently toggle the appearance of the string between bold and not-bold by pressing CTRL-B; independently toggle between italicized and non-italicized appearance by pressing CTRL-I.; and independently toggle the appearance of the string between underlined and not underlined by pressing CTRL-U. Pressing the left mouse button once on a selected string makes it unselected.

Ans:



Mistake in each state, transition, or concurrency = -1 mark

--- The End---