

Advanced Programming (CSE201), Quiz -1

Time allocated: 03:20pm – 3:45pm (25 minutes) + 5 minutes for uploading solution

Instructions:

- You must follow all the instructions sent to you earlier over the email.
- Only reasonable and clearly mentioned assumptions (if any) would be accepted.
- For justifications, please be as concise as possible (2-3 sentences only)

Q1) Read the below mentioned problem description and answer below questions:

“A vehicle shop repairs the engine of Bikes and Cars and replaces the windshield of Trucks and Cars. This shop has a mechanic responsible for receiving the Bikes, Cars, and Trucks; storing them in two different garages inside the vehicle shop; and for providing the services. These garages are essentially storing bikes, cars, and trucks according to the requested service. One of the garages can store bikes and cars to repair their engine, whereas the other can store trucks and cars for replacing the windshield. For repairing the engine, the bike and car must be turned off. Turning off the bike makes it save its gps location, whereas turning off the car simply switches off its ignition. For replacing the windshield, the clean engine mode must be activated on cars and trucks. Clean engine mode in the car saves its gps location whereas in the truck it will switch off the radio connection with the truck company. The mechanic receives the bike, car, and truck throughout the day and then services them together by taking them out of their respective garages. Any bike, car, and truck can be given for only one type of service.”

Implement the Object Oriented Implementation of the above program description and identify the class relationships (if any). No need to code the main method. You should only use the concepts taught in Lectures up to interfaces and polymorphism. You must write actual code (no pseudocode). **[7 marks]**

Rubric for Q1: (Total 7 marks)

No need to see the working code. Only see the below points:

- 1) Classes Shop and Employee
 - a) +0.5 marks
- 2) Data encapsulation being followed inside each classes Shop and Employee
 - a) +0.5 marks
- 3) Two List (or ArrayList) type objects inside Shop class as List<InterfaceA> and List<InterfaceB> (Polymorphism in parameter)
 - a) +0.5 marks
- 4) If Shop instantiates Employee then composition relationship, else association relationship / If employee inherits the shop class
 - a) +0.75 marks

- 5) The employee has association relationship with Shop / If the employee has protected access of shop class variables or has their access through a getter function
a) +0.5 marks
- 6) InterfaceA/Abstract_classA with method declaration "public void serviceA()"
a) +0.5 marks
- 7) InterfaceB/Abstract_classB with method declaration "public void serviceB()"
a) +0.5 marks
- 8) Three things come for service (classes TypeX, TypeY, and TypeZ)
a) +0.75 marks
- 9) class TypeX implements/extends InterfaceA/Abstract_classA and provide concrete implementation of serviceA
a) +0.5 marks
- 10) Class TypeY implements/extends InterfaceA/Abstract_classA and InterfaceB/Abstract_classB and provide concrete implementations of methods serviceA and serviceB
a) +1 marks
- 11) Class TypeZ implements/extends InterfaceB/Abstract_classB and provide concrete implementation of serviceB
a) +0.5 marks
- 12) All overridden methods annotated with @Override
a) +0.5 marks

Q2) Correct the following program. It must follows all principles/coding practices of OOP [3 marks]

```
public class Main{
    public static void main(String[] args){
        String name passed here correctly
        i) +0.75 marks

        */
        Bird B1 = new Eagle("); Eagle("abc")
        Bird B2 = new Bird();
        /* Eagle on both LHS and RHS: +0.75 marks */
        Eagle B2 = new Eagle("def");
        B1.eat();
        B1.fly(B2);
    }
}

public interface Bird{
    public void eat();
    private public void fly(Bird Eagle B); // 0.25 + 0.5 marks
}
```

```
public class Eagle implements Bird {  
    private String name;  
    public Eagle(String n){name = n; }  
  
    public void eat(){  
        System.out.println("Eagle Eating");  
    }  
  
    public void fly(Bird Eagle B){ // +0.75 marks  
        System.out.println(this.name+" flying with " +B.name);  
    }  
}
```