

XXY car dealership owners try a new model for employee engagement. They have a point system to encourage their employees. Each employee is awarded **10** points for a sold car.

With these points, employees can achieve one of four status levels: excellent, good, satisfactory, or sufficient. The level will determine the salary bonus to which they are entitled as well as other benefits.



The total number of points collected during the current month will determine which of the four status levels they are assigned for the following month: For example only the points collected in September will determine the status level and bonus for October.

The Points class keeps details of the points and status levels of each employee.

```
public class Points{
    private String name; // name of the employee
    private String employeeNo; // id of the employee
    private int totalPoints; // this month's points
    private String statusNow; // current(this month's)status: excellent, good,
                                // satisfactory, or sufficient
    private String statusNextMonth; // following month's status
    private Sales[] allSales = new Sales[31]; //details of the sales
                                                // during this month

    int d; // number of days with sales this month
    public Points(String id) { // constructor for new employee
        this(id, "Salesperson X");
    }
                                //constructor for new employee given the name
    public Points(String id, String n){
        name=n;
        employeeNo = id;
        totalPoints = 0;
        d = 0;
        statusNow = "sufficient";
    }
    // all the accessor and mutator methods are present but not shown

    public Sales getAllSales(int day){
        return allSales[day];
    }
    public void addSales(Sales s){ // adds a new Sales object to the array
        allSales[d] = s;
        d = d + 1;
    }
    isExcellent() {code missing}

    calculateTotalPoints() {code missing}

    carsMissing() {code missing}
}
```

1. The instance variables in the Points class are preceded by the modifier private. The choice of modifier affects the way in which these variables are accessed or used.

(a) With the use of two examples other than private, outline how the choice of this modifier affects the way in which these variables are accessed or used. [4]

(b) With reference to the two methods with the same name in the Points class, explain the OOP feature that makes it possible to successfully implement either of these methods. [4]

The employees will be assigned one of four levels for the following month (**Excellent, good, satisfactory, or sufficient**) depending upon the current month's total points as follows.

- **sufficient** = less than 30 points
- **satisfactory** = 30 or more but less than 50
- **good** = 50 or more but less than 70
- **excellent** = 70 or more.

In September, Bob became a new employee. So far, in September, Bob has sold five cars.

(c) State the status level that Bob has been assigned, for September. [1]

The different Points objects are stored in an array which is declared globally in the main (driver) class as follows: `Points[] allPoints = new Points[10000];`

(d) State how an individual object can be identified using this array. [1]

The attribute `statusNow` is assigned its correct value at the beginning of every month. It cannot be changed during the month.

(e) Construct the method `isExcellent()` in the Points class, which will return whether the current status is "Excellent". [3]

2. The details of sales during the current months are stored in the variable `allSales` which is an array of the Sales class. `allSales` is used in determining the total points awarded in the current month. The employee can sell only 5 cars per day.

The Sales class is outlined below:

```
public class Sales{
    private String carCode[]; // id of the sold cars
    private double price[]; // price of the sold cars
    private int numOfCars; // to store number of cars sold
    public Sales(){
        carCode = new String[5];
        price = new double[5];
        numOfCars=0;
    }
    public Sales(String carID, double price){
        this();
        addACar(String carID, double price); // you will create this method
    }
}
```

```

    public int getNumOfCars() {
        return numOfCars;
    }
}

```

(a) Create method `addAcar(String carID, double price)` that will add a sold car to a sale and update points. [3]

(b) Construct a UML diagram for the Sales class. [3]

The main (driver) class manages the Points and Sales classes. It contains the following code:

```

Points[] allPoints = new Points[10000];           // declared globally
allPoints[0] = new Points("em3435");
allPoints[1] = new Points("em3355", "Kim");
allPoints[2] = new Points("em7766", "Anna");
Sales s1 = new Sales ();
s1.addAcar("Civic123", 35000.0);
s1.addAcar("Juke345", 28000.0);
Sales s2 = new Sales ();
s2.addAcar("Corolla774", 25500.0);

Sales s3 = new Sales ("BMW11", 100000.0);
allPoints[0].addSales(s1);
allPoints[0].addSales(s2);
allPoints[1].addSales(s3);
allPoints[1].addSales(new Sales ("HRV44", 85500.0));

```

(c) State the output given by the following statements:

(i) `System.out.println(allPoints[2].getEmployeeNo());` [1]

(ii) `System.out.println(allPoints[0].getName());` [1]

(iii) `System.out.println(allPoints[1].getAllsales(1).getNumOfCars());` [1]

(d) Construct the method `calculateTotalPoints()`, in the Points class, which will calculate and return the total number of points awarded so far in the current month. [5]

(e) Construct the method `carsMissing()`, in the Points class, that will return the number of extra cars that an employee needs to sell in order to keep the same status the following month as they have in the current month. [7]

(f) what is the relationship between Points and Sales classes? [1]

The car dealership maintains the details of the extra benefits that each status provides in appropriate classes.

(g) Suggest how the car dealership might make use of the inheritance feature of OOP when designing the classes from pages 1 and 2. [4]

At the end of the month at midnight the systems are temporarily made unavailable as the dealership makes the changes needed to prepare the system for the new month.

(h) Identify three changes that would have to be made to the classes previously described in order for the system to function correctly in the new month. [3]

3. Full details of each car are stored as objects of the Car class.

For research purposes, the managers want to identify the name of the car that a particular employee has sold the most during the current month.

Without writing code, outline the steps that would have to be taken to accomplish this.

You should include reference to any classes, methods or attributes that will be created or will be needed. [7]

4. Full details of the employee are stored as objects of the Employee class. This class is partially shown below:

```
public class Employee{
    private String employeeNo;
    private String email; //email address (assume only 1 per employee)

    public Employee(String a, String b){
        employeeNo = a;
        email = b;
    }

    public String getEmployeeNo () {
        return employeeNo;
    }
    public String getEmail()
    {
        return email;
    }
}
```

The objects can be accessed through the ArrayList allEmployees which is declared in the main (driver) class as follows:

```
ArrayList<Employee> allEmployees = new ArrayList<Employee>();
```

Construct the method excellentMails() that will return an ArrayList containing the email addresses of all current "Excellent" employees. You should make use of any previously defined methods. [6]