Option D — Object-oriented programming

A hotel chain has a loyalty scheme in which customers are awarded 1000 points for each day they stay in one of their hotels. With these points, customers can achieve one of three status levels: Gold, Silver or Bronze. The level will determine the extra services to which they are entitled.

The total number of points collected during the **current** year will determine which of the three status levels they are assigned for the **following** year: For example **only** the points collected in 2018 will determine the status level for 2019.

Occasionally, new customers receive additional **bonus** points as part of a promotion.

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

```
public class Points
{
  private String memberId;  // id of the hotel customer
  private int totalPoints;  // this year's points
  private int bonusPoints;  // any bonus points given to this year's new member
  private String statusNow; // current(this year's)status
  private String statusNextYear; // following year's status
  private Visits[] allVisits = new Visits[366];//details of each visit
                                                // during this year
  int y;                    // number of visits this year

  public Points(String id) // constructor for new member
  {
    memberId = id;
    bonusPoints = 0;
    y = 0;
    statusNow = "Bronze";
  }

  //constructor for new member given bonus points (valid for current year only)
  public Points(String id, int bp)
  {
    memberId = id;
    bonusPoints = bp; // multiples of 1000 - maximum number is 5000
    y = 0;
    statusNow = "Bronze";
  }

  // all the accessor and mutator methods are present but not shown

  public Visits getAllVisits(int v)
  {
    return allVisits[v];
  }

  public void addVisit(Visits v) // adds a new Visit object to the array
  {
    allVisits[y] = v;
    y = y + 1;
  }

  isGold()   {code missing}
  calculateTotalPoints(){code missing}
  daysMissing(){code missing}
}
```

(Option D continues on the following page)

(Option D continued)

10. 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 customers will be assigned one of three levels for the following year (Gold, Silver or Bronze) depending upon the current year's total points as follows.

- Bronze = less than 10 000 points
- Silver = 10 000 or more but less than 50 000
- Gold = 50 000 or more.

In 2018, Tim became a member for the first time and was awarded a bonus of 1000 points. So far, in 2018, Tim has stayed three times at one of these hotels. The first visit lasted 2 days, the second visit lasted 1 day and the third visit lasted 6 days.

(c) State the status level that Tim has been assigned, for 2019, following these visits. [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 year for existing members. It cannot be changed during the year.

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

(Option D continues on the following page)

(Option D continued)

11. The details of hotel stays during the current year are stored in the variable `allVisits` which is an array of the Visits class. `allVisits` is used in determining the total points awarded in the current year.

The `Visits` class is outlined below:

```
public class Visits
{
  private String hotelCode;  //  id of the hotel
  private int days;          //  number of days of the visit

  public Visits(String h, int d)
  {
    hotelCode = h;
    days = d;
  }

  public int getDays()
  {
    return days;
  }
}
```

(a) Construct a UML diagram for the `Visits` class. [3]

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

```
Points[] allPoints = new Points[10000]; // declared globally

allPoints[0] = new Points("m100");
allPoints[1] = new Points("m101",5000);
allPoints[2] = new Points("m102",2000);

Visits v1 = new Visits("h003", 3);
Visits v2 = new Visits("h013", 1);
Visits v3 = new Visits("h013", 2);
Visits v4 = new Visits("h005", 6);

allPoints[0].addVisit(v1);
allPoints[0].addVisit(v2);
allPoints[0].addVisit(v3);
allPoints[0].addVisit(v4);
allPoints[1].addVisit(v1);
allPoints[1].addVisit(new Visits("h004",6));
```

(Option D continues on the following page)

(Option D, question 11 continued)

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

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

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

(iii) `System.out.println(allPoints[1].getAllVisits(1).getDays());` [1]

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

(d) Construct the method `daysMissing()`, in the `Points` class, that will return the number of extra days that a customer needs to stay in order to keep the same status the following year as they have in the current year. [7]

The hotel chain maintains the details of the extra benefits that each status provides in appropriate classes.

(e) Suggest how the hotel chain might make use of the inheritance feature of OOP when designing the classes from pages 12 and 14. [4]

At midnight on 31st December each year the systems are temporarily made unavailable as the hotel chain makes the changes needed to prepare the system for the new year.

(f) 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 year. [3]

12. Full details of each hotel are stored as objects of the `Hotel` class.

For research purposes, the managers want to identify the name of the hotel in which a particular customer has stayed the most days during the current year.

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]

End of Option D