COMP 248

OBJECT ORIENTED PROGRAMMING

Tutorial #11: Arrays of Primitive Type Review & Arrays of Objects - SOLUTIONS

Question 1:

What will be the output of the following code?

A.

```
class Parray1
{
   public static void main(String[] args)
   {
      int i;
      int a[] = {5, 2, 3, 1, 1, 0, 2, 1, 0, 1};
      for (i = 0; (i < 10); i++)
      {
         if (a[i] == 0)
            break;
         if (i % 3 == 0)
            continue;
         System.out.println("a[" + i + "]:" + a[i]);
      }
   }
}
```

Answer:

```
a[1]:2
a[2]:3
a[4]:1
```

B.

```
class Parray2
{
   public static void main(String[] args)
   {
      int[] data = {1,3,5,8,11,15};
      int sum = 0;
      for(int i = 1; i < data.length; ++i)
      {
         sum = sum + data[i] - data[i-1];
         System.out.println("sum = " + sum);
      }
   }
}
```

Answer:

```
sum = 2
sum = 4
sum = 7
sum = 10
sum = 14
```

C.

```
class Parray3
{
   public static int sumIf (int[] a, boolean[] b)
   {
      int sum = 0;
      for (int i = 0; i < a.length; ++i)
         if(b[i])
            sum = sum + a[i];
      return sum;
   }

   public static void main(String[] args)
   {
      int[] data = {1, 2, 3, 4, 5, 6, 7};
      boolean[] filter = {true, false, true, true, false, true, true};
      System.out.println("data:" + sumIf(data, filter));
      for(int i = 0; i < filter.length; ++i)
         filter[i] = !filter[i];
         System.out.println("data:" + sumIf(data, filter));
   }
}
```

Answer:

```
data:21
data:7
```

Question 2:

Write a static method called sum2 that has two parameters called row and n. The parameter row is an array of numbers of type double; n is an integer which will be greater than or equal to 0. The method will return the sum of the first n elements of the array row.

Answer:

```
public static double sum2 (double [] row, int n)
{
    if (n > row.length)
    {
          n = row.length; // to avoid an Index-Out-Of-Bounds error
    }
    double sum = 0;
    for (int i = 0; i < n; i++)
    {
          sum += row[i];
    }
    return sum;
}
```

Question 3:

Assume the following class:

```
public class Airplane {
      private int nbOfPassengers;
      private double weight;
      private int maxSpeed;

       . . .
}
```

A.

Complete the class Airplane by adding

- A default constructor
- A constructor with 2 parameters which will hold data for the `weight` and `maxSpeed` attributes.
- A constructor with 3 parameters.
- All accessor and mutator methods
- An equals method which returns `true` if all attributes are the same and `false` otherwise
- A `toString` such that it returns the string

```
Plane with capacity of <nbOfPassengers> passengers, weighing <weight>kg can travel
at a maximum speed of <maxSpeed>km/hr.
```

where `<nbOfPassengers>` , `<weight>` and `<maxSpeed>` are the values stored in the attributes of an object.

```
Answer:
public class Airplane {
    private int nbOfPassengers;
    private double weight;
    private int maxSpeed;

    public Airplane() {

    }

    public Airplane(double weight, int maxSpeed) {
          this.weight = weight;
          this.maxSpeed = maxSpeed;
    }
```

```
   public Airplane(int nbOfPassengers, double weight, int maxSpeed) {
         this.nbOfPassengers = nbOfPassengers;
         this.weight = weight;
         this.maxSpeed = maxSpeed;
   }

   public int getNbOfPassengers() {
         return nbOfPassengers;
   }

   public void setNbOfPassengers(int nbOfPassengers) {
         this.nbOfPassengers = nbOfPassengers;
   }

   public double getWeight() {
         return weight;
   }

   public void setWeight(double weight) {
         this.weight = weight;
   }

   public int getMaxSpeed() {
         return maxSpeed;
   }

   public void setMaxSpeed(int maxSpeed) {
         this.maxSpeed = maxSpeed;
   }

   public boolean equals(Airplane other) {
         return maxSpeed == other.maxSpeed &&
                    nbOfPassengers == other.nbOfPassengers &&
                    weight == other.weight;
   }

   public String toString() {
         return "Plane with capacity of " + nbOfPassengers +
         " passengers, weighing "
   + weight + "kg, can travel at a maximum speed of " + maxSpeed + "km/hr.";
   }
}
```

B.

Write a driver class which:

- Contains a static method called `getAverageWeight()` that takes an array of `Airplane` objects as parameter and computes and returns the average weight of the airplanes in the array.

  Answer:

  ```
  public static double getAverageWeight(Airplane[] a) {
        double sum = 0;
        for (int i = 0; i < a.length; ++i)
              sum += a[i].getWeight();
        if (a.length != 0)
              return (sum/a.length);
        else
              return 0;

  }
  ```

- Contains a static method called `findFasterAirplane()` that takes an array of `Airplane` objects as parameter and returns the fastest airplane in the array. If several airplanes have the same maximum speed, return null.

  Answer:

  ```
  public static Airplane findFasterAirplane(Airplane[] a) {
        if (a == null)
              return null;
        Airplane fastestSoFar = a[0];
        for (int i = 1; i < a.length; ++i) {
              if (a[i].getMaxSpeed() > fastestSoFar.getMaxSpeed())
                    fastestSoFar = a[i];
              if (a[i].getMaxSpeed() == fastestSoFar.getMaxSpeed())
                    return null;
        }
        return fastestSoFar;
  }
  ```

- Contains a `main()` method which will create an array capable of holding 5 `Airplane` objects using each constructor at least once. Prompt the user for the data for the first 3 objects. For the 4th object ask only for the weight and maximum speed. For the last object in the array use the default constructor.
- Display the content of the array of `Airplane` objects.

- The object that was created with the default constructor, should be assigned attribute values from 3 different objects in the array. (You decide which one).
- Using the `findFasterAirplane()` method find the fastest airplane and display its information.
- Using the `getAverageWeight()` method find the average weight of the airplanes in the array and display its information.
- Compare any 2 objects in the array to see if they are equal using the *equals* method.
- Compare any 2 objects in the array to see if they are equal using the accessor methods (not the equals() methods)
- Have 2 objects contain the same data and then using the *equals* method compare them.

Answer:

```
import java.util.Scanner;
public class AirplaneTester {

     public static void main(String[] args) {
          Scanner keyIn = new Scanner(System.in);
          Airplane [] fleet = new Airplane[5];

          /* will create an array of Airplane objects using each constructor at
           * least once. Prompt the user for the data for each object.
           * (5 objects will be enough).
           */
          System.out.println("Enter info for 3 planes in this order\n" +
               "number of passengers, weight and maximum speed:");
          for (int i = 0; i < 3; i++)
          {
               System.out.print("Plane #" + (i + 1) + ": ");
               int nbP = keyIn.nextInt();
               double weight = keyIn.nextDouble();
               int mxSp = keyIn.nextInt();
               fleet[i] = new Airplane(nbP, weight, mxSp);
          }
          System.out.println("\nEnter info for 1 plane in this order\n" +
               "weigth   maximum speed:");
          double weight = keyIn.nextDouble();
          int mxSp = keyIn.nextInt();
          fleet[3] = new Airplane(weight, mxSp);
          fleet[4] = new Airplane();
```

```
//Display the content of the array of Airplane objects.
System.out.println("\nHere are the Airplane objects created:");
for (int i = 0; i < fleet.length; i++)
      System.out.println(fleet[i]);

/* The object that was created with the default constructor,
 * should be assigned attribute values from 3 different objects in the array.
 * (You decide which one).
 */
fleet[4].setNbOfPassengers(fleet[0].getNbOfPassengers());
fleet[4].setWeight(fleet[1].getWeight());
fleet[4].setMaxSpeed(fleet[2].getMaxSpeed());

/* Using the findFasterAirplane() method find the fastest
 * airplane and display its information.
 */
System.out.println("\nThe fastest airplane is: " +
      findFasterAirplane(fleet));

/* Using the getAverageWeight() method find the average weight of
 *  the airplanes in the array and display its information.
 */
System.out.println("\nThe average weight of the airplanes is: " +
      getAverageWeight(fleet) + "kg");

/* Compare any 2 objects in the array to see if they are equal using
 * the equals method.
 */
if(fleet[0].equals(fleet[4]))
      System.out.println("Same data in 2 objects");
else
      System.out.println("Not same data in 2 objects");

/* Compare any 2 objects in the array to see if they are
 * equal using the accessor methods (not the equals() methods)
 */
if(fleet[0].getNbOfPassengers()== fleet[2].getNbOfPassengers() &&
      fleet[0].getWeight() == fleet[2].getWeight() &&
      fleet[0].getMaxSpeed() == fleet[2].getMaxSpeed())
      System.out.println("Same data in 2 objects");
else
      System.out.println("Not same data in 2 objects");
```

```java
        /* Have 2 objects contain the same data and then using the equals
         * method compare them.
         */
        fleet[0].setNbOfPassengers(fleet[2].getNbOfPassengers());
        fleet[0].setWeight(fleet[2].getWeight());
        fleet[0].setMaxSpeed(fleet[2].getMaxSpeed());
        if(fleet[0].equals(fleet[2]))
            System.out.println("Same data in 2 objects");
        else
            System.out.println("Not same data in 2 objects");
        keyIn.close();
    }

    //-------------------------------------
    // Static Methods
    //-------------------------------------

    public static double getAverageWeight(Airplane[] a)
    {
        double sum = 0;
        for (int i = 0; i < a.length; ++i) sum += a[i].getWeight();
        if (a.length != 0)
            return (sum/a.length);
        else
            return 0;
    }

    public static Airplane findFasterAirplane(Airplane[] a)
    {
        if (a == null)
            return null;
        Airplane fastestSoFar = a[0];
        for (int i = 1; i < a.length; ++i)
        {
          if (a[i].getMaxSpeed() == fastestSoFar.getMaxSpeed()) return null;

          if (a[i].getMaxSpeed() > fastestSoFar.getMaxSpeed()) fastestSoFar = a[i];
        }
        return fastestSoFar;
    }
}
```