Question 1 - In the following array, what is the value of `fees.length`?

```
double[][] fees={{3.00, 3.50},{6.35, 7.35},{9.00, 5.00}};
```

a. 1
b. 2
c. 4
d. 6
e. None of the above

Question 2 - Having multiple class methods of the same name where each method has a different number of or type of parameters is known as:

a. method overriding
b. method overloading
c. information hiding
d. encapsulation
e. None of the above

Question 3 - What is *y* after the following switch statement?

```
int x=65, y=66, z=67;
switch (z-x)  {
  case 1: y = 67; break;
  case 2: y = 68; break;
  default: y = 69;
}
System.out.print(y);
```

A. 65
B. 67
C. 68
D. 69
E. None of the above

Question 4 - The following nested loop structure will execute the inner most statement (x++) how many times?

```
for (int j = 0; j < 100; j++)
     for (int k = 10; k > 0; k--)
          x++;
```

a. 100
b. 200
c. 1,000
d. 2,000
e. None of the above

Question 5 - In Java, arrays are ____________

a. primitive data types
b. primitive data types if the type stored in the array is a primitive data type and objects otherwise
c. objects
d. interfaces
e. None of the above

Question 6 - Assume an int array, *candy*, stores the number of candy bars sold by a group of children where *candy[j]* is the number of candy bars sold by child *j*. Assume there are 12 children in all.
What does the following method do?

```
public int question6( ){
     int value1 = 0;
     int value2 = 0;
     for (int j = 0; j < 12; j++)
          if (candy[j] > value1){
               value1 = candy[j];
               value2 = j;
          }
     return value2;
}
```

a. It returns the total number of children who sold 0 candy bars
b. It returns the index of the child who sold the most candy bars
c. It returns the number of candy bars sold by the child who sold the most candy bars
d. It returns the total number of children who sold more than 0 candy bars
e. None of the above

Question 7 – Static methods can

a. invoke other static methods
b. reference any instance data
c. reference non-static instance data
d. invoke non-static methods
e. None of the above

Question 8 - The *Container* class has one integer attribute named *area* that is set to a default value of 3. There is one accessor method named *getArea()* that returns the area.
Choose the code to fill in that will display the following on the screen: 333

```
Container [] c = {new Container(), new Container(), new Container()};
for(int i = 0; i < c.length; i++)
    System.out.print(______code to fill in______);
```

a. c.area
b. c[i].getArea()
c. c[area]
d. c[i]. area
e. None of the above

Question 9 - In the following code, what is the printout?

```
public class Test {
  public static void main(String[] args) {
       int[] list1 = {1, 0, 1};
       int[] list2 = {0, 1, 0};
       list2 = list1;
       list1[0] = 0; list1[1] = 1; list2[2] = 2;
       for (int i = 0; i < list1.length; i++)
            System.out.print(list1[i] + " ");
  }
}
```

a. 1 2 1
b. 0 1 2
c. 0 1 1
d. 1 0 1
e. None of the above

Question 10 - To initialize a String array names to store the three Strings "Huey", "Duey" and "Louie", you would do

a. String names = {"Huey", "Duey", "Louie"};
b. String[ ] names = new String{"Huey", "Duey", "Louie"};
c. String names[3] = {"Huey", "Duey", "Louie"};
d. String names; names[0] = "Huey"; names[1] = "Duey"; names[2] = "Louie";
e. None of the above

Question 11 (20 points)

Given the class `Point` below, define a class `Circle` that represent a circle with a given center and radius. The `Circle` class should have a center attribute named **center** as well as a floating point **radius** attribute. The center is a point object, defined by the class `Point`. The class should also have these members:

- the **constructor** of the class, which should take **parameters** to initialize **all attributes**
- a getter for **center**
- a setter for **radius**
- a method **liesOnCircle()** that takes a point parameter and returns true if the point lies on the circumference of the circle and false otherwise.

```
public class Point
{
   private int x, y;

   public Point(int x, int y)
   {
      this.x=x;
      this.y=y;
    }

    public String toString()
    {
      return "x:"+x+","+"y:"+y;
    }

    public double distance(Point other)
    {
      return Math.sqrt(Math.pow(x-other.x,2)+Math.pow(y-other.y,2));
    }
}
```

Question 12 (20 points)

Write an application CircleTest that creates and populates an array of 50 circles with center and radius data from the user. The program calculates and prints to the screen the number of circles that contain origin (0, 0) on their circumference.