1. Given the following java statement:

```
Toys[] arr = new Toys[12];
```

Which statement declares and initializes variable *n* to the index of the last element of the array?

a. `int n = Toys.length();`
b. `int n = arr.length() - 1;`
c. `int n = arr.length();`
d. `int n = arr.length - 1;`
e. None of the above

2. What is the output?

```
char letter = 'A';
switch (letter + 2) {
   case 'B': letter +=1; break;
   case 'C': letter +=2;
   default: letter +=1;
}
System.out.print(letter);
```

a. C
b. B
c. D
d. Syntax error
e. None of the above

3. Which two Java methods are overloaded methods?

a. `public int add(int a, int b)` and `public int add(int c, int d)`
b. `public int add(int a, int b)` and `public float add(int c, int d)`
c. `public int add(int a, int b)` and `public int add(int a, float b)`
d. `public float add(float a, float b)` and `public float add(float c, float d)`
e. None of the above

4. What is the output of the following code?

```
int[] arr = {1, 2, 3, 4, 5};
for (int i : arr){
    System.out.print(arr[i]);
    i--;
 }
```

a. `12345`
b. `01234`
c. `1234`
d. `234Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5`
e. None of the above

5. What is the output?

```
public static void replace(int[] allGrades, int examScore) {
    int temp=allGrades[0];
    allGrades[0] = examScore;
    examScore = temp;

}
public static void main(String args[]) {
     int[] myGrades = {72,84,75,95};
     replace(myGrades,  myGrades[3]);
     System.out.println(myGrades[0]+", "+myGrades[3]);
     }
```

a. 95, 75
b. 95, 95
c. 95, 72
d. 72, 95
e. None of the above

6. The_____case defined in a recursive method stops recursion.

a base
b general
c recursive
d loop
e None of the above

7. Which statement declares and instantiates a two-dimensional integer array called myArray with 4 rows and 2 columns?

a `int myArray[4][2];`
b `int[] myArray = new int[4];`
c `int[][] myArray = new int (4)(2);`
d `int[][] myArray = new int[2][4];`
e None of the above

8. Given a class with a static variable xVar, and a collection of objects of that class. Which of the following statement is always true?

a Each object will have its own instance of xVar
b Changing xVar in one object will alter xVar for all the other objects
c Changing xVar in one object will only alter xVar in that objects
d Objects do not share xVar
e None of the above

9. If the recursive method below is invoked as recPrint(3), what is the output?

```
public static void recPrint(int n) {
         if (n == 0) {
            System.out.print("!");
         }
         else {
            recPrint(n-1);
            System.out.print(n);
         }
     }
```

a. !123
b. 123!
c. 321!
d. !321
e. none of the above

10. Given the class Shoe. Choose the code to fill in that will correctly compare shoes in order to find the position of the shoe with the largest side.

```
class Shoe{
   private int size;
   public Shoe(int size){
      this.size=size;
    }
   public int getSize(){ return size;}
}

class TestShoe{
   public static void main(String[] args)
   {
      Shoe[] shoes={new Shoe(7), new Shoe(9),new Shoe(5)};
      int maxPos = 0;
      for (int i=1; i<shoes.length; i++)
         if (_____code to fill in_____))
            maxPos = i;
      System.out.println(maxPos);
}
```

a shoes[i].size > shoes[maxPos].size
b shoes[i] > shoes[maxPos]
c shoes[i].getSize() > shoes[maxPos].getSize()
d shoes[maxPos].getSize() > shoes[i].getSize()
e None of the above

Question 11

The class GPA below defines the gpa of a student in terms of its value. The value of the GPA is a floating point number between 0.0 and 4.0.

Write a class **`Student`**, that represents a student with a given name and gpa. The Student class should have two attributes: name and gpa objects. The gpa of a student is defined by the class GPA. The class should also have these members:

- A **constructor** that take parameters to initialize the Student object
- A getter for **gpa**
- A setter for **name**
- A method **honorRoll()** that takes a gpa object parameter and returns true if the student's gpa value is greater than or equal to the parameter gpa value and false otherwise
- A **toString()** method returning the name of the students and his/her gpa value.

```
public class GPA
{
   private double value;

   public GPA(double value)
   { this.value =value;}

   public double getValue()
   {  return value;}

   public int compareTo(GPA other){
      if (this.value <other.value)
         return -1;
      else if (this.value==other.value)
         return 0;
      else
         return 1;
   }
   public String toString()
   {  return Double.toString(value);}
}
```

Question 12
A family consisting of several members, each with their own cell phone is analyzing the cellphone bills received over a number of months.

Write a method `bestInMonths()` that takes one parameter:

1. `famCellBills`, a two-dimensional array of doubles, where each row represents a member of the family and the columns represent the months included in the analysis

The method returns **an array** containing the lowest bill received in each month over all members of the family. For example, the following table illustrate a sample of an analysis, where Best Bill represents the best (lowest) amounts received by a family member for each month:

	Jan	Feb	March
Jane	**34.56**	75.34	23.87
Jack	87.79	**32.05**	23.54
Sophie	89.78	66.76	87.99
Kylie	45.34	56.45	**16.54**
Best Bill	34.56	32.05	16.54