Day-7
Quiz-2

1. Create a class illustrating all the three types of constructors
● No arguments constructor
● Default constructor
● Parameterised constructor (can create more than one with different type of parameters)

CODE:

```
public class ConstructorsExample {
    public ConstructorsExample() {
        System.out.println("No arguments constructor called");
    }
    public ConstructorsExample(int parameter1) {
        System.out.println("Parameterized constructor with one int parameter called.
Parameter1: " + parameter1);
    }
    public ConstructorsExample(String parameter1) {
        System.out.println("Parameterized constructor with one String parameter called.
Parameter1: " + parameter1);
    }
    public ConstructorsExample(String parameter1, int parameter2) {
        System.out.println("Parameterized constructor with String and int parameters called.
Parameter1: " + parameter1 + ", Parameter2: " + parameter2);
    }
    public static void main(String[] args) {
        ConstructorsExample obj1 = new ConstructorsExample();
        ConstructorsExample obj2 = new ConstructorsExample(10);
constructor with one int parameter
        ConstructorsExample obj3 = new ConstructorsExample("Hello");
constructor with one String parameter
        ConstructorsExample obj4 = new ConstructorsExample("World", 20);
constructor with String and int parameters
    }
}
```

OUTPUT:

```
No arguments constructor called
Parameterized constructor with one int parameter called. Parameter1: 10
Parameterized constructor with one String parameter called. Parameter1:
    HelloParameterized constructor with String and int parameters called. Parameter1:
    World, Parameter2: 20
```

2. Given a sorted integer array (in increasing order), remove duplicates in-place such that each unique element appears only once. The relative order of the elements should be kept the same. Then return the number of unique elements in the array.

Input
[22,22,77,77,88, 89,89]

Output
4
Explanation : After removing duplicates -> [22, 77, 88, 89, _, _, _]
No. of unique elements = 4

CODE:

```
import java.util.Arrays;
import java.util.Scanner;
public class RemoveDuplicates {
   public static int removeDuplicates(int[] nums) {
      if (nums == null || nums.length == 0) {
         return 0;
      }
      int uniqueCount = 1; // At least one element is unique
      for (int i = 1; i < nums.length; i++) {
         if (nums[i] != nums[i - 1]) {
            nums[uniqueCount] = nums[i];
            uniqueCount++;
         }
      }
      return uniqueCount;
   }
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the number of elements in the array: ");
      int n = scanner.nextInt();
      int[] inputArray = new int[n];
      System.out.println("Enter the elements in increasing order:");
      for (int i = 0; i < n; i++) {
         inputArray[i] = scanner.nextInt();
      }
      int result = removeDuplicates(inputArray);
      System.out.println("After removing duplicates -> " + Arrays.toString(inputArray));
      System.out.println("No. of unique elements = " + result);
      scanner.close();
   }
}
```

OUTPUT:

```
Enter the number of elements in the array: 7
Enter the elements in increasing order:
22
22
77
77
88
89
89
After removing duplicates -> [22, 77, 88, 89, 88, 89, 89]
No. of unique elements = 4
```

3 . An array contains both positive and negative numbers in random order. Rearrange the array elements so that all negative numbers appear before all positive numbers. Don't use .sort() method

Input [-12, 11, -13, -5, 6, -7, 5, -3, -6]
Output [-12, -13, -5, -7, -3, -6, 11, 6, 5]

CODE:

```
import java.util.Arrays;
import java.util.Scanner;
public class RearrangeArray {
    public static void rearrangeArray(int[] nums) {
        int n = nums.length;
        int negativeIndex = 0;
        for (int i = 0; i < n; i++) {
            if (nums[i] < 0) {
                int temp = nums[i];
                nums[i] = nums[negativeIndex];
                nums[negativeIndex] = temp;

                negativeIndex++;
            }
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] inputArray = new int[n];
        System.out.println("Enter the elements in random order:");
```

```
        for (int i = 0; i < n; i++) {
            inputArray[i] = scanner.nextInt();
        }
        System.out.println("Input: " + Arrays.toString(inputArray));
        rearrangeArray(inputArray);
        System.out.println("Output: " + Arrays.toString(inputArray));
        scanner.close();
    }
}
```

OUTPUT:

```
Enter the number of elements in the array: 9
Enter the elements in random order:
-12
11
-13
-5
6
-7
5
-3
-6
Input: [-12, 11, -13, -5, 6, -7, 5, -3, -6]
Output: [-12, -13, -5, -7, -3, -6, 5, 6, 11]
```