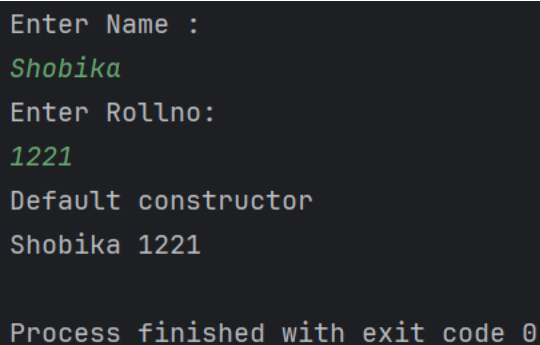


1. Create a class illustrating all the three types of constructors

- No arguments constructor
- Default constructor
- Parameterised constructor (can create more than one with different type of parameters)

```
import java.util.*;
public class Main
{
    public Main()
    {
        System.out.println("Default constructor");
    }
    public Main(String name ,int rollno)
    {
        System.out.println(name+" "+rollno);
    }
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        String name ;
        int rollno;
        System.out.println("Enter Name :");
        name = scan.nextLine();
        System.out.println("Enter Rollno:");
        rollno = scan.nextInt();
        Main defaultcons = new Main();
        Main parametercons = new Main(name,rollno);

    }
}
```

A screenshot of a terminal window showing the execution of a Java program. The user is prompted to enter a name and a roll number. The name 'Shobika' is entered, and the roll number '1221' is entered. The program then prints 'Default constructor' and 'Shobika 1221'. Finally, it shows 'Process finished with exit code 0'.

```
Enter Name :
Shobika
Enter Rollno:
1221
Default constructor
Shobika 1221

Process finished with exit code 0
```

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

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static int removeDuplicates(int[] nums) {
```

```
        if (nums == null || nums.length == 0) {
```

```
            return 0;
```

```
        }
```

```
        int uniqueCount = 1;
```

```
        int n = nums.length;
```

```
        for (int i = 1; i < n; 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 size of the array: ");
```

```
        int size = scanner.nextInt();
```

```
        int[] nums = new int[size];
```

```
        System.out.println("Enter the sorted array elements:");
```

```
        for (int i = 0; i < size; i++) {
```

```
            nums[i] = scanner.nextInt();
```

```
        }
```

```
        scanner.close();
```

```
        int uniqueCount = removeDuplicates(nums);
```

```
        System.out.println("No. of unique elements = " + uniqueCount);
```

```

        System.out.print("Modified array with unique elements: ");
        for (int i = 0; i < uniqueCount; i++) {
            System.out.print(nums[i] + " ");
        }
    }
}

```

```

Enter the size of the array: 7
Enter the sorted array elements:
22 22 77 77 88 89 89
No. of unique elements = 4
Modified array with unique elements: 22 77 88 89
Process finished with exit code 0

```

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]

```

import java.util.*;
public class Main {
    public static void rearrangeArray(int[] nums) {
        if (nums == null || nums.length <= 1) {
            return;
        }

        int n = nums.length;

        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                if (nums[i] >= 0 && nums[j] < 0) {

                    int temp = nums[i];
                    nums[i] = nums[j];
                    nums[j] = temp;
                }
            }
        }
    }

    public static void main(String[] args) {
        int[] nums = {-12, 11, -13, -5, 6, -7, 5, -3, -6};

        System.out.println("Original array: " + Arrays.toString(nums));

        rearrangeArray(nums);

        System.out.println("Rearranged array: " + Arrays.toString(nums));
    }
}

```

```
Original array: [-12, 11, -13, -5, 6, -7, 5, -3, -6]  
Rearranged array: [-12, -13, -5, -7, -3, -6, 5, 6, 11]  
  
Process finished with exit code 0
```