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:**

import java.util.\*;

public class Constructor {

public Constructor() {

System.out.println("No arguments constructor.");

}

public Constructor(int value) {

System.out.println("Default constructor. Value: " + value);

}

public Constructor(String name, int age) {

System.out.println("Parameterized constructor. Name: " + name + ", Age: " + age);

}

public static void main(String[] args) {

Constructor obj1 = new ConstructorExample();

Constructor obj2 = new ConstructorExample(19);

Constructor obj3 = new ConstructorExample("Serra", 27);

}

}

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 Main {

public static int removeDuplicates(int[] nums) {

if (nums.length == 0) {

return 0;

}

int uniqueCount = 1;

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 size of an array: ");

int size = scanner.nextInt();

int[] nums = new int[size];

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

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

nums[i] = scanner.nextInt();

}

scanner.close();

int uniqueCount = removeDuplicates(nums);

System.out.println("Number of unique elements: " + uniqueCount);

}

}

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.Scanner;

public class Main {

public static void rearrangeArray(int[] arr) {

int n = arr.length;

int negativeIndex = 0;

for (int i = 0; i < n; i++) {

if (arr[i] < 0) {

int temp = arr[i];

arr[i] = arr[negativeIndex];

arr[negativeIndex] = temp;

negativeIndex++;

}

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the size of an array: ");

int size = scanner.nextInt();

int[] arr = new int[size];

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

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

arr[i] = scanner.nextInt();

}

rearrangeArray(arr);

System.out.println("Array after rearranging: " + java.util.Arrays.toString(arr));

}

}