Assignment no:-6

Declare a single-dimensional array of 5 integers inside the main method. Traverse the array to print the default values. Then accept records from the user and print the updated values of the array

**package** org.example;

**import** java.util.Scanner;

**public** **class** Arrayexample {

**public** **static** **void** main(String[] args) {

**int**[]numbers = **new** **int**[5];

System.***out***.println("Default values of the array");

**for** (**int** index = 0; index<=5; ++index ) {

System.***out***.println("arr [index]");

System.***out***.println("\nEnter 5 integers to update the array:");

**for** (**int** i = 0; i < numbers.length; i++) {

System.***out***.print("Enter value for index " + i + ": ");

**while** (!sc.hasNextInt()) {

System.***out***.println("Invalid input. Please enter an integer.");

sc.next();

System.***out***.print("Enter value for index " + i + ": ");

}

numbers[i] = sc.nextInt();

System.***out***.println("\nUpdated values of the array:");

**for** ( i = 0; i<= 5; ++i) {

System.***out***.println("Index " + i + ": " + numbers[i]);

sc.close();

}

}

}

}

}

Output: Default values of the array

arr [index]

Enter 5 integers to update the array:

Enter value for index 0:

1. Write a program to find the maximum and minimum values in a single-dimensional array of integers.

**package** program14;

**public** **class** MaxMinFinder {

**public** **static** **int**[] findMaxMin(**int**[] arr) {

**if** (arr.length == 0) {

**throw** **new** IllegalArgumentException("Array is empty");

}

**int** maxVal = arr[0];

**int** minVal = arr[0];

**for** (**int** num : arr) {

**if** (num > maxVal) {

maxVal = num;

}

**if** (num < minVal) {

}

minVal = num;

}

**return** **new** **int**[]{maxVal,minVal};

**public** **static** **void** main(String[] args) {

**int**[] array = {34, -23, 0, 12, 78, -56, 19};

**try** {

**int**[] result = *findMaxMin*(array);

**int** maxVal = result[0];

**int** minVal = result[1];

System.***out***.println("Maximum value: " + maxVal);

System.***out***.println("Minimum value: " + minVal);

} **catch** (IllegalArgumentException e) {

System.***out***.println(e.getMessage());

}

}

}

Output: Maximum value: 78

Minimum value: -56

writea program to remove duplicate elements from a single-dimensional array of integers.

Solu: **package** program14;

**import** java.util.HashSet;

**import** java.util.Arrays;

**public** **class** RemoveDuplicates {

// Method to remove duplicates from an array

**public** **static** **int**[] removeDuplicates(**int**[] arr) {

HashSet<Integer> set = **new** HashSet<>();

**for** (**int** num : arr) {

set.add(num);

}

**int**[] result = **new** **int**[set.size()];

**int** index = 0;

**for** (**int** num : set) {

result[index++] = num;

}

**return** result;

}

**public** **static** **void** main(String[] args) {

**int**[] array = {34, 12, 78, 34, -56, 12, 19, -56};

**int**[] uniqueArray = *removeDuplicates*(array);

System.***out***.println("Array without duplicates: " + Arrays.*toString*(uniqueArray));

}

}

Output: Array without duplicates: [34, 19, -56, 12, 78]

**package** program14;

**import** java.util.HashSet;

**import** java.util.Set;

**import** java.util.Arrays;

**public** **class** ArrayIntersection {

**public** **static** **int**[] findIntersection(**int**[] arr1, **int**[] arr2) {

Set<Integer> set1 = **new** HashSet<>();

**for** (**int** num : arr1) {

set1.add(num);

}

Write a program to find the intersection of two single-dimensional arrays.

Set<Integer> intersection = **new** HashSet<>();

**for** (**int** num : arr2) {

**if** (set1.contains(num)) {

intersection.add(num);

}

}

**int**[] result = **new** **int**[intersection.size()];

**int** index = 0;

**for** (**int** num : intersection) {

result[index++] = num;

}

**return** result;

}

**public** **static** **void** main(String[] args) {

**int**[] array1 = {1, 3, 5, 7, 9};

**int**[] array2 = {3, 6, 9, 12};

**int**[] intersection = *findIntersection*(array1, array2);

System.***out***.println("Intersection of the two arrays: " + Arrays.*toString*(intersection));

}

}

1. Write a program to find the missing number in an array of integers ranging from 1 to N.

Sol:

**package** program14;

**public** **class** MissingNumberFinder {

**public** **static** **int** findMissingNumber(**int**[] arr, **int** N) {

**int** expectedSum = N \* (N + 1) / 2;

**int** actualSum = 0;

**for** (**int** num : arr) {

actualSum += num;

}

**return** expectedSum - actualSum;

}

**public** **static** **void** main(String[] args) {

**int**[] array = {1, 2, 4, 5, 6};

**int** N = 6;

**int** missingNumber = *findMissingNumber*(array, N);

System.***out***.println("The missing number is: " + missingNumber);

}

}

Output:

The missing number is: 3

Modify the previous assignment to use getter and setter methods instead of acceptRecord and printRecord.

Sol:

**package** program14;

**public** **class** Record {

**private** String name;

**private** **int** age;

**public** **void** acceptRecord(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**public** **void** printRecord() {

System.***out***.println("Name: " + name);

System.***out***.println("Age: " + age);

}

**public** **static** **void** main(String[] args) {

Record record = **new** Record();

record.acceptRecord("Alice", 30);

record.printRecord();

}

}

Output :Name: Alice

Age: 30