1. 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 in.cadac.Assignment6.Question1;**

**import java.util.Scanner;**

**public class singledimensionalarray {**

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

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

**Scanner sc=new Scanner(System.*in*);**

**System.*out*.println("Default Values of Array");**

**for(int row = 0; row < arr.length; ++ row )**

**{**

**System.*out*.println("Array element at index value "+ row +":"+arr[row ] );**

**}**

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

**System.*out*.println("Enter Values of Array");**

**for( int row = 0; row < arr.length; ++ row )**

**{**

**arr[row] =sc.nextInt();**

**}**

**System.*out*.println("Default Values of Array");**

**for( int row = 0; row < arr.length; ++ row )**

**{**

**System.*out*.println("Array element at index value "+ row +":"+arr[row ] );**

**}**

**sc.close();**

**}**

**}**

1. Declare a single-dimensional array of 5 integers inside the main method. Define a method named acceptRecord to get input from the terminal into the array and another method named printRecord to print the state of the array to the terminal.

**package** in.cdac.Assignment6.Question2;

**import** java.util.Scanner;

**public** **class** AcceptrecordArray {

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

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

*acceptRecord*(array);

*printRecord*(array);

}

**public** **static** **void** acceptRecord(**int**[] array) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.println("Enter 5 integers:");

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

array[i] = scanner.nextInt();

}

}

**public** **static** **void** printRecord(**int**[] array) {

System.***out***.println("Array elements:");

**for** (**int** element : array) {

System.***out***.println(element);

}

}

}

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

package Min\_Max\_Array;

import java.util.Scanner;

public class Minarray {

public static void main(String[] args) {

int[] array = new int[5];

acceptRecord(array);

printRecord(array);

}

public static void acceptRecord(int[] array) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter 5 integers:");

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

array[i] = scanner.nextInt();

}

}

public static void printRecord(int[] array) {

System.out.println("Array elements:");

for (int element : array) {

System.out.print(" "+element);

}

System.out.println(" ");

int max = findMax(array);

int min = findMin(array);

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

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

}

public static int findMax(int[] array) {

int max = array[0];

for (int i = 1; i < array.length; i++) {

if (array[i] > max) {

max = array[i];

}

}

return max;

}

public static int findMin(int[] array) {

int min = array[0];

for (int i = 1; i < array.length; i++) {

if (array[i] < min) {

min = array[i];

}

}

return min;

}

}

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

package Duplicate;

import java.util.Scanner;

public class Duplicate {

public static void main(String[] args) {

int[] array = new int[5];

Scanner sc = new Scanner(System.in);

System.out.println("Enter 5 integers:");

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

array[i] = sc.nextInt();

}

array = removeDuplicates(array);

System.out.println("Array after removing duplicates:");

for (int element : array) {

if (element != 0) {

System.out.println(element);

}

}

sc.close();

}

public static int[] removeDuplicates(int[] array) {

int[] temp = new int[array.length];

int j = 0;

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

int count = 0;

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

if (array[i] == array[k]) {

count++;

}

}

if (count == 0) {

temp[j++] = array[i];

}

}

for (int i = 0; i < array.length; i++) { // Replace original array with temp array

array[i] = temp[i];

}

return array;

}

}

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

package in.cdac.intersectionarray;

import java.util.Scanner;

public class arrayIntersection{

public static void main(String[] args) {

int[] array = new int[5];

int[] array2 = new int[5];

acceptRecord(array, array2);

printRecord(array, array2);

intersectionarray(array, array2);

}

public static void acceptRecord(int[] array, int[] array2) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter 5 integers:");

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

array[i] = scanner.nextInt();

}

System.out.println("Enter 5 integers:");

for (int j = 0; j < array2.length; j++) {

array2[j] = scanner.nextInt();

}

}

public static void printRecord(int[] array, int[] array2) {

System.out.println("Array elements:");

for (int element : array) {

System.out.println(element);

System.out.println("Array elements:");

for ( int element2 : array2) {

System.out.println(element2);

}

}

}

public static void intersectionarray(int[] array, int[] array2) {

System.out.println("Intersection of two Array ");

int count = 0;

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

for (int j = 0; j < array2.length; j++) {

if (array[i] == array2[j]) {

count++;

break;

}

}

}

int[] intersection = new int[count];

int index = 0;

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

for (int j = 0; j < array2.length; j++) {

if (array[i] == array2[j]) {

intersection[index++] = array[i];

break;

}

}

}

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

System.out.println(intersection[i]);

}

}

}

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

**package** in.cdac.Missingnumber;

**import** java.util.Scanner;

**public** **class** Missingnum {

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

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

**int** sum = 0;

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

sum += num;

}

**return** expectedSum - sum;

}

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

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the value of N (the range): ");

**int** N = sc.nextInt();

**int**[] arr = **new** **int**[N - 1];

System.***out***.println("Enter " + (N - 1) + " integers:");

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

arr[i] = sc.nextInt();

}

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

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

sc.close();

}

}

1. Declare a single-dimensional array as a field inside a class and instantiate it inside the class constructor. Define methods named acceptRecord and printRecord within the class and test their functionality.

package in.cdac.singledimensionarray;

import java.util.Scanner;

class Array {

private int[] arr;

public Array(int size) {

arr = new int[size]; // Instantiate the array

}

public void acceptRecord() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter " + arr.length + " integers:");

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

arr[i] = sc.nextInt();

}

sc.close();

}

public void printRecord() {

System.out.println("Array elements are:");

for (int num : arr) {

System.out.print(num + " ");

}

System.out.println();

}

}

public class SingleDimensionalArray {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int size = sc.nextInt();

Array a= new Array(size);

a.acceptRecord();

a.printRecord();

sc.close();

}

}

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

**package** in.cdac.singledimensionarray;

**import** java.util.Scanner;

**class** singleArray {

**private** **int**[] arr;

**public** singleArray(**int** size) {

arr = **new** **int**[size];

}

**public** **void** setArrayValues() {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.println("Enter " + arr.length + " integers:");

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

arr[i] = scanner.nextInt();

}

scanner.close();

}

**public** **int**[] getArrayValues() {

**return** arr;

}

**public** **int** getArray() {

System.***out***.println("Array elements are:");

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

System.***out***.print(num + " ");

}

**return** 0;

}

}

**public** **class** SingleDimensionalArray {

**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();

singleArray sa = **new** singleArray (size);

sa.setArrayValues();

sa.getArray();

scanner.close();

}

}

1. You need to implement a system to manage airplane seat assignments. The airplane has seats arranged in rows and columns. Implement functionalities to:

* Initialize the seating arrangement with a given number of rows and columns.
* Book a seat to mark it as occupied.
* Cancel a booking to mark a seat as available.
* Check seat availability to determine if a specific seat is available.
* Display the current seating chart.

**package** in.cdac.Airplane;

**public** **class** Test {

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

System.***out***.println("Welcome to the Airplane Seat Management System!");

**int** rows = seatManagerutil.*getInput*("Enter number of rows: ");

**int** columns = seatManagerutil.*getInput*("Enter number of columns: ");

seatManager manager = **new** seatManager(rows, columns);

**boolean** exit = **false**;

**while** (!exit) {

seatManagerutil.*displayMenu*();

**int** choice = seatManagerutil.*getInput*("");

**switch** (choice) {

**case** 1: // Book a seat

**int** bookRow = seatManagerutil.*getInput*("Enter row to book: ");

**int** bookCol = seatManagerutil.*getInput*("Enter column to book: ");

**if** (manager.bookSeat(bookRow, bookCol)) {

System.***out***.println("Seat booked successfully.");

} **else** {

System.***out***.println("Seat already booked or invalid seat.");

}

**break**;

**case** 2: // Cancel a booking

**int** cancelRow = seatManagerutil.*getInput*("Enter row to cancel: ");

**int** cancelCol = seatManagerutil.*getInput*("Enter column to cancel: ");

**if** (manager.cancelSeat(cancelRow, cancelCol)) {

System.***out***.println("Booking canceled successfully.");

} **else** {

System.***out***.println("No booking found or invalid seat.");

}

**break**;

**case** 3: // Check seat availability

**int** checkRow = seatManagerutil.*getInput*("Enter row to check: ");

**int** checkCol = seatManagerutil.*getInput*("Enter column to check: ");

**if** (manager.isSeatAvailable(checkRow, checkCol)) {

System.***out***.println("Seat is available.");

} **else** {

System.***out***.println("Seat is not available.");

}

**break**;

**case** 4: // Display seating chart

manager.displaySeats();

**break**;

**case** 5: // Exit

exit = **true**;

System.***out***.println("Exiting system.");

**break**;

**default**:

System.***out***.println("Invalid option! Please try again.");

}

}

}

}