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** com.assign;

**import** java.util.Scanner;

**public** **class** A6\_Q1 {

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

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

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

**for**(**int** i :a)

{

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

}

System.***out***.println("Enter values for array:");

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

{

a[i]=sc.nextInt();

}

**for**(**int** i :a)

{

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

}

}

}

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** com.assign;

**import** java.util.Scanner;

**public** **class** A6\_Q2 {

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

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

System.***out***.println("Enter values for array:");

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

{

a[i]=*sc*.nextInt();

}

}

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

**for**(**int** i :a)

{

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

}

}

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

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

*acceptRecord*(a);

*printRecord*(a);

}

}

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

**package** com.assign;

**import** java.util.Scanner;

**public** **class** A6\_Q3 {

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

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

System.***out***.println("Enter values for array:");

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

{

a[i]=*sc*.nextInt();

}

}

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

**for**(**int** i :a)

{

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

}

}

**public** **static** **void** maxMin(**int**[] a) {

**int** max=a[0];

**for**(**int** i=1;i<a.length;i++)

{

**if**(a[i]>max) {

max=a[i];

}}

System.***out***.println("Max value is "+max);

**int** min=a[0];

**for**(**int** i=1;i<a.length;i++)

{

**if**(a[i]<min) {

min=a[i];

}

}

System.***out***.println("Min value is "+min);

}

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

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

*acceptRecord*(a);

*maxMin*(a);

}

}

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

import java.util.Scanner;

public class RemoveDuplicatesManual {

// Method to remove duplicates using nested loops

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

int n = array.length;

int[] temp = new int[n];

int index = 0;

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

boolean isDuplicate = false;

// Check if the element is a duplicate

for (int j = 0; j < index; j++) {

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

isDuplicate = true;

break;

}

}

// If not a duplicate, add to temp array

if (!isDuplicate) {

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

}

}

// Copy the non-duplicate elements to a new array

int[] result = new int[index];

System.arraycopy(temp, 0, result, 0, index);

return result;

}

// Method to print the contents of an array

public static void printArray(int[] array) {

for (int num : array) {

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

}

System.out.println();

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Ask user for the size of the array

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

int size = sc.nextInt();

int[] array = new int[size];

// Accept input for the array

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

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

array[i] = sc.nextInt();

}

// Remove duplicates and print the result

int[] uniqueArray = removeDuplicates(array);

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

printArray(uniqueArray);

}

}

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

**package** com.assign;

**public** **class** A6\_Q5 {

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

// Define two arrays

**int**[] array1 = {1, 2, 3, 4, 5};

**int**[] array2 = {3, 4, 5, 6, 7};

// Find the intersection of the two arrays

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

// Print the intersection

System.***out***.print("Intersection of the arrays: ");

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

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

}

}

**public** **static** **int**[] findIntersection(**int**[] array1, **int**[] array2) {

// Create an array to store the intersection elements

**int**[] temp = **new** **int**[Math.*min*(array1.length, array2.length)];

**int** count = 0;

// Iterate through each element in array1

**for** (**int** i = 0; i < array1.length; i++) {

// Check if this element is in array2

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

**if** (array1[i] == array2[j]) {

// Check if the element is already added to the temp array

**boolean** alreadyExists = **false**;

**for** (**int** k = 0; k < count; k++) {

**if** (temp[k] == array1[i]) {

alreadyExists = **true**;

**break**;

}

}

// If not already added, add it to the temp array

**if** (!alreadyExists) {

temp[count++] = array1[i];

}

**break**; // Element found, no need to check further

}

}

}

// Create the final intersection array with the exact size

**int**[] intersection = **new** **int**[count];

**for** (**int** i = 0; i < count; i++) {

intersection[i] = temp[i];

}

**return** intersection;

}

}

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

**package** com.assign;

**import** java.util.Scanner;

**public** **class** A6\_Q6 {

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

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

System.***out***.println("Enter values for array:");

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

{

a[i]=*sc*.nextInt();

}

}

**public** **static** **void** missedElement(**int**[] a) {

**for**(**int** i=1;i<=a.length;i++)

{ **int** count=0;

**for**(**int** j=1;j<a.length;j++)

{**if**(i==a[j-1])

count++;

}

**if**(count==0)

System.***out***.println("Missing element:" +i);

}

}

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

**for**(**int** i :a)

{

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

}

}

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

System.***out***.println("Enter no. of elements in an array:");

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

**int**[] a= **new** **int**[n];

*acceptRecord*(a);

*missedElement*(a);

}

}

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** com.assign;

**import** java.util.Scanner;

**public** **class** A6\_Q7 {

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

**private** **int**[] a;

A6\_Q7(**int** n)

{

a=**new** **int**[n];

}

**public** **void** acceptRecord(**int** n) {

System.***out***.println("Enter values for array:");

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

{

a[i]=*sc*.nextInt();

}

}

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

**for**(**int** i :a)

{

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

}

}

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

A6\_Q7 a=**new** A6\_Q7(5);

a.acceptRecord(5);

a.printRecord();

}

}

1. Modify the previous assignment to use getter and setter methods instead of acceptRecord and printRecord.
2. 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** com.assign;

**public** **class** A6\_Q9 {

**private** **boolean**[][] seats;

// Constructor to initialize the seating arrangement with a given number of rows and columns

**public** A6\_Q9(**int** rows, **int** columns) {

seats = **new** **boolean**[rows][columns];

}

// Method to book a seat

**public** **void** bookSeat(**int** row, **int** column) {

**if** (isSeatAvailable(row, column)) {

seats[row][column] = **true**;

System.***out***.println("Seat booked at (" + row + ", " + column + ").");

} **else** {

System.***out***.println("Seat (" + row + ", " + column + ") is already booked.");

}

}

// Method to cancel a booking

**public** **void** cancelBooking(**int** row, **int** column) {

**if** (!isSeatAvailable(row, column)) {

seats[row][column] = **false**;

System.***out***.println("Booking canceled for seat (" + row + ", " + column + ").");

} **else** {

System.***out***.println("Seat (" + row + ", " + column + ") is already available.");

}

}

// Method to check if a specific seat is available

**public** **boolean** isSeatAvailable(**int** row, **int** column) {

**return** !seats[row][column];

}

// Method to display the current seating chart

**public** **void** displaySeatingChart() {

System.***out***.println("Seating Chart:");

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

**for** (**int** j = 0; j < seats[i].length; j++) {

**if** (seats[i][j]) {

System.***out***.print("X "); // Occupied seat

} **else** {

System.***out***.print("O "); // Available seat

}

}

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

}

}

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

// Initialize the airplane seating with 5 rows and 4 columns

A6\_Q9 airplaneSeating = **new** A6\_Q9(5, 4);

// Display the initial seating chart

airplaneSeating.displaySeatingChart();

// Book a few seats

airplaneSeating.bookSeat(0, 0);

airplaneSeating.bookSeat(2, 3);

airplaneSeating.bookSeat(4, 1);

// Display the seating chart after booking

airplaneSeating.displaySeatingChart();

// Try to book an already booked seat

airplaneSeating.bookSeat(0, 0);

// Cancel a booking

airplaneSeating.cancelBooking(2, 3);

// Display the seating chart after canceling

airplaneSeating.displaySeatingChart();

}

}