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.

import java.util.Scanner;

public class ArrayTraversal {

public static void main(String[] args) {

int[] numbers = new int[5];

// Print default values

System.out.println("Default values:");

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

System.out.print(numbers[i] + " ");

}

System.out.println();

// Accept records from the user

Scanner scanner = new Scanner(System.in);

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

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

numbers[i] = scanner.nextInt();

}

// Print updated values

System.out.println("Updated values:");

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

System.out.print(numbers[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.

import java.util.Scanner;

public class ArrayMethods {

public static void acceptRecord(int[] numbers) {

Scanner scanner = new Scanner(System.in);

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

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

numbers[i] = scanner.nextInt();

}

}

public static void printRecord(int[] numbers) {

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

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

System.out.print(numbers[i] + " ");

}

System.out.println();

}

public static void main(String[] args) {

int[] numbers = new int[5];

acceptRecord(numbers);

printRecord(numbers);

}

}

1. Write a program to find the maximum and minimum values in a single-dimensional array of integers.
2. Write a program to remove duplicate elements from a single-dimensional array of integers.
3. Write a program to find the intersection of two single-dimensional arrays.

import java.util.Arrays;

public class ArrayIntersection {

public static void main(String[] args) {

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

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

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

int count = 0;

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

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

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

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

break;

}

}

}

// Resize the intersection array to the actual count

intersection = Arrays.copyOf(intersection, count);

System.out.println("Intersection:");

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

System.out.print(intersection[i] + " ");

}

}

}

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

public class MissingNumber {

public static void main(String[] args) {

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

int n = numbers.length + 1; // Assuming numbers range from 1 to N

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

int actualSum = 0;

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

actualSum += numbers[i];

}

int missingNumber = expectedSum - actualSum;

System.out.println("Missing number: " + missingNumber);

}

}

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.
2. **mport** java.util.Scanner;
3. **public** **class** MyArrayClass {
4. **private** **int**[] numbers;
5. **public** MyArrayClass(**int** size) {
6. numbers = **new** **int**[size];
7. }
8. **public** **void** acceptRecord() {
9. Scanner scanner = **new** Scanner(System.***in***);
10. System.***out***.println("Enter " + numbers.length + " integers:");
11. **for** (**int** i = 0; i < numbers.length; i++) {
12. numbers[i] = scanner.nextInt();
13. }
14. }
15. **public** **void** printRecord() {
16. System.***out***.println("Array elements:");
17. **for** (**int** i = 0; i < numbers.length; i++) {
18. System.***out***.print(numbers[i] + " ");
19. }
20. System.***out***.println();
21. }
22. **public** **static** **void** main(String[] args) {
23. MyArrayClass myArray = **new** MyArrayClass(5);
24. myArray.acceptRecord();
25. myArray.printRecord();
27. }
29. }
30. Modify the previous assignment to use getter and setter methods instead of acceptRecord and printRecord.

**package** com.jbk;

**public** **class** GandS {

**private** **int**[] numbers;

**public** GandS(**int** size) {

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

}

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

**return** numbers;

}

**public** **void** setNumbers(**int**[] numbers) {

**this**.numbers = numbers;

}

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

GandS myArray = **new** GandS(5);

myArray.getNumbers()[10] = 10; // Accessing and modifying using getter and setter

((Object) myArray).printRecord();

}

}

}

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.

import java.util.Scanner;

public class AirplaneSeatAssignment {

private int rows;

private int columns;

private boolean[][] seatingChart;

public AirplaneSeatAssignment(int rows, int columns) {

this.rows = rows;

this.columns = columns;

seatingChart = new boolean[rows][columns];

}

public void bookSeat(int row, int column) {

if (isSeatAvailable(row, column)) {

seatingChart[row - 1][column - 1] = true;

System.out.println("Seat " + row + " " + column + " booked successfully.");

} else {

System.out.println("Seat " + row + " " + column + " is already occupied.");

}

}

public void cancelBooking(int row, int column) {

if (isSeatOccupied(row, column)) {

seatingChart[row - 1][column - 1] = false;

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

} else {

System.out.println("Seat " + row + " " + column + " is already empty.");

}

}

public boolean isSeatAvailable(int row, int column) {

return !seatingChart[row - 1][column - 1];

}

public boolean isSeatOccupied(int row, int column) {

return seatingChart[row - 1][column - 1];

}

public void displaySeatingChart() {

System.out.println(" ");

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

System.out.print(" " + (i + 1) + " ");

}

System.out.println();

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

System.out.print((i + 1) + " ");

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

System.out.print(seatingChart[i][j] ? "X " : " ");

}

System.out.println();

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of rows: ");

int rows = scanner.nextInt();

System.out.print("Enter the number of columns: ");

int columns = scanner.nextInt();

AirplaneSeatAssignment airplane = new AirplaneSeatAssignment(rows, columns);

while (true) {

System.out.println("1. Book a seat");

System.out.println("2. Cancel a booking");

System.out.println("3. Check seat availability");

System.out.println("4. Display seating chart");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.print("Enter the row number: ");

int row = scanner.nextInt();

System.out.print("Enter the column number: ");

int column = scanner.nextInt();

airplane.bookSeat(row, column);

break;

