**Name:** Yash Bandu Dhole

**Centre:** Juhu.

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.

**Code:**

package problem1;

import java.util.Scanner;

public class singleDArray {

public static void main(String[] args) {

int[] arr = new int[5];

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

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

System.***out***.println("arr[" + i + "] = " + arr[i]);

}

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

System.***out***.println("\nEnter 5 integer values:");

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

System.***out***.print("Enter value for arr[" + i + "]: ");

arr[i] = scanner.nextInt();

}

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

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

System.***out***.println("arr[" + i + "] = " + arr[i]);

}

scanner.close();

}

}

**O/p:**

Default values of the array:

arr[0] = 0

arr[1] = 0

arr[2] = 0

arr[3] = 0

arr[4] = 0

Enter 5 integer values:

Enter value for arr[0]: 1

Enter value for arr[1]: 2

Enter value for arr[2]: 3

Enter value for arr[3]: 4

Enter value for arr[4]: 5

Updated values of the array:

arr[0] = 1

arr[1] = 2

arr[2] = 3

arr[3] = 4

arr[4] = 5

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.

**Code:**

package problem2;

import java.util.Scanner;

public class modularArray {

public static void main(String[] args) {

int[] arr = new int[5];

*acceptRecord*(arr);

*printRecord*(arr);

}

public static void acceptRecord(int[] arr) {

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

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

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

System.***out***.print("Enter value for arr[" + i + "]: ");

arr[i] = scanner.nextInt();

}

}

public static void printRecord(int[] arr) {

System.***out***.println("\nValues in the array:");

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

System.***out***.println("arr[" + i + "] = " + arr[i]);

}

}

}

**O/p:**

Enter 5 integer values:

Enter value for arr[0]: 1

Enter value for arr[1]: 2

Enter value for arr[2]: 3

Enter value for arr[3]: 4

Enter value for arr[4]: 5

Values in the array:

arr[0] = 1

arr[1] = 2

arr[2] = 3

arr[3] = 4

arr[4] = 5

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

**Code:**

package problem3;

import java.util.Scanner;

public class maxMinArray {

public static void main(String[] args) {

int [] arr = new int[5];

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

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

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

System.***out***.println("Enter value for arr[" +i+ "]:");

arr[i]=sc.nextInt();

}

int MAX =arr[0];

int MIN = arr[0];

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

if (arr[i]>MAX) {

MAX=arr[i];

}

if(arr[i]<MIN) {

MIN=arr[i];

}

}

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

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

sc.close();

}

}

**O/p:**

Enter 5 integer values:

Enter value for arr[0]:

8

Enter value for arr[1]:

1

Enter value for arr[2]:

5

Enter value for arr[3]:

4

Enter value for arr[4]:

9

Max value is: 9

Min value is: 1

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

**Code:**

package problem4;

import java.util.Scanner;

public class duplicateArray {

public static void main(String[] args) {

int[] arr = new int[7];

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

System.***out***.println("Enter 7 integer values:");

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

arr[i] = scanner.nextInt();

}

int n = arr.length;

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

for (int j = i + 1; j < n; j++) {

if (arr[i] == arr[j]) {

for (int k = j; k < n - 1; k++) {

arr[k] = arr[k + 1];

}

n--;

j--;

}

}

}

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

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

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

}

scanner.close();

}

}

**O/p:**

Enter 7 integer values:

8

7

5

8

9

6

7

Array after removing duplicates:

8 7 5 9 6

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

**Code:**

package problem5;

import java.util.Scanner;

public class arrayIntersection {

public static void main(String[] args) {

int[] arr1 = new int[5];

int[] arr2 = new int[5];

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

System.***out***.println("Enter 5 elements for the first array:");

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

arr1[i] = scanner.nextInt();

}

System.***out***.println("Enter 5 elements for the second array:");

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

arr2[i] = scanner.nextInt();

}

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

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

for (int j = 0; j < 5; j++) {

if (arr1[i] == arr2[j]) {

System.***out***.print(arr1[i] + " ");

break;

}

}

}

}

}

**O/p:**

Enter 5 elements for the first array:

7

4

8

5

9

Enter 5 elements for the second array:

1

2

3

4

5

Intersection of the two arrays:

4 5

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

**Code:**

package problem6;

import java.util.Scanner;

public class missingNumber {

public static void main(String[] args) {

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

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

int N = scanner.nextInt();

int[] arr = new int[N - 1]; // Array with one number missing

System.***out***.println("Enter " + (N - 1) + " elements from 1 to " + N + ":");

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

arr[i] = scanner.nextInt();

}

int sumN = N \* (N + 1) / 2; // Sum of numbers from 1 to N

int sumArray = 0;

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

sumArray += arr[i];

}

int missingNumber = sumN - sumArray;

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

}

}

**O/p:**

Enter the value of N: 6

Enter 5 elements from 1 to 6:

1

2

5

6

3

The missing number is: 4

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.

**Code:**

package problem7;

import java.util.Iterator;

import java.util.Scanner;

class testArrayHandler{

private int[] arr;

public testArrayHandler(int size) {

arr = new int[size];

}

public void acceptRecord() {

Scanner scanner = new Scanner(System.in);

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

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

System.out.println("Enter value for arr[" + i + "]: ");

arr[i] = scanner.nextInt();

}

scanner.close();

}

public void printRecord() {

System.out.println("Values in the array: ");

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

System.out.println("arr [" + i + "] =" + arr[i]);

}

}

}

public class arrayHandler {

public static void main(String[] args) {

testArrayHandler handler = new testArrayHandler(5);

handler.acceptRecord();

handler.printRecord();

}

}

**O/p:**

Enter 5 integer values:

Enter value for arr[0]:

8

Enter value for arr[1]:

7

Enter value for arr[2]:

3

Enter value for arr[3]:

4

Enter value for arr[4]:

6

Values in the array:

arr [0] =8

arr [1] =7

arr [2] =3

arr [3] =4

arr [4] =6

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

**Code:**

package problem8;

import java.util.Scanner;

class ArrayHandler{

private int[] arr;

public ArrayHandler(int size) {

arr = new int[size];

}

public void setValue(int index, int val) {

if(index >=0 && index < arr.length) {

arr[index] =val;

}else {

System.***out***.println(" Index out of bounds ");

}

}

public int getValue(int index) {

if(index >= 0 && index<arr.length) {

return arr[index];

}else {

System.***out***.println("Index out of bounds");

return -1;

}

}

public int[] getArray() {

return arr;

}

}

public class getsetArray {

public static void main(String[] args) {

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

ArrayHandler handler = new ArrayHandler(5);

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

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

System.***out***.println("Enter value for arr[" + i + "]: ");

handler.setValue(i, scanner.nextInt());

}

System.***out***.println("Values in the array: ");

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

System.***out***.println("arr[" + i +"] = " + handler.getValue(i));

}

}

}

**O/p:**

Enter 5 integer values:

Enter value for arr[0]:

5

Enter value for arr[1]:

4

Enter value for arr[2]:

3

Enter value for arr[3]:

6

Enter value for arr[4]:

9

Values in the array:

arr[0] = 5

arr[1] = 4

arr[2] = 3

arr[3] = 6

arr[4] = 9

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.

**Code:**

package problem9;

import java.util.Scanner;

class AirplaneSeating {

private boolean[][] seats; // 2D array representing seating arrangement (true = booked, false = available)

// Constructor to initialize seating arrangement with given rows and columns

public AirplaneSeating(int rows, int columns) {

seats = new boolean[rows][columns];

}

// Method to book a seat

public void bookSeat(int row, int col) {

if (row < seats.length && col < seats[0].length) {

if (!seats[row][col]) {

seats[row][col] = true;

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

} else {

System.***out***.println("Seat is already booked.");

}

} else {

System.***out***.println("Invalid seat selection.");

}

}

// Method to cancel a booking

public void cancelBooking(int row, int col) {

if (row < seats.length && col < seats[0].length) {

if (seats[row][col]) {

seats[row][col] = false;

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

} else {

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

}

} else {

System.***out***.println("Invalid seat selection.");

}

}

// Method to check if a seat is available

public boolean isSeatAvailable(int row, int col) {

if (row < seats.length && col < seats[0].length) {

return !seats[row][col];

}

System.***out***.println("Invalid seat selection.");

return false;

}

// Method to display the seating chart

public void displaySeatingChart() {

System.***out***.println("Current seating chart:");

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

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

System.***out***.print(seats[i][j] ? "[X] " : "[O] ");

}

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

}

}

}

public class AirplaneSeatManager {

public static void main(String[] args) {

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

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

int rows = scanner.nextInt();

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

int columns = scanner.nextInt();

AirplaneSeating airplane = new AirplaneSeating(rows, columns);

while (true) {

System.***out***.println("\n1. 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("Choose an option: ");

int option = scanner.nextInt();

if (option == 1) {

System.***out***.print("Enter row to book: ");

int row = scanner.nextInt();

System.***out***.print("Enter column to book: ");

int col = scanner.nextInt();

airplane.bookSeat(row - 1, col - 1); // Subtracting 1 to account for 0-based indexing

} else if (option == 2) {

System.***out***.print("Enter row to cancel: ");

int row = scanner.nextInt();

System.***out***.print("Enter column to cancel: ");

int col = scanner.nextInt();

airplane.cancelBooking(row - 1, col - 1);

} else if (option == 3) {

System.***out***.print("Enter row to check: ");

int row = scanner.nextInt();

System.***out***.print("Enter column to check: ");

int col = scanner.nextInt();

if (airplane.isSeatAvailable(row - 1, col - 1)) {

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

} else {

System.***out***.println("Seat is already booked.");

}

} else if (option == 4) {

airplane.displaySeatingChart();

} else if (option == 5) {

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

break;

} else {

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

}

}

}

}

**o/p:**

Enter number of rows: 10

Enter number of columns: 7

1. Book a seat

2. Cancel a booking

3. Check seat availability

4. Display seating chart

5. Exit

Choose an option: 4

Current seating chart:

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

1. Book a seat

2. Cancel a booking

3. Check seat availability

4. Display seating chart

5. Exit

Choose an option: 1

Enter row to book: 1

Enter column to book: 1

Seat booked successfully.

1. Book a seat

2. Cancel a booking

3. Check seat availability

4. Display seating chart

5. Exit

Choose an option: 4

Current seating chart:

[X] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

[O] [O] [O] [O] [O] [O] [O]

1. Book a seat

2. Cancel a booking

3. Check seat availability

4. Display seating chart

5. Exit

Choose an option: 3

Enter row to check: 2

Enter column to check: 3

Seat is available.

1. Book a seat

2. Cancel a booking

3. Check seat availability

4. Display seating chart

5. Exit

Choose an option: 5

Exiting system.