***Assignment – 6***

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.

Ans = package array\_assign;

import java.util.Iterator;

public class Q1 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int arr[]=new int[5];

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

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

}

}

}

package array\_assign;

import java.util.Iterator;

import java.util.Scanner;

public class Q1 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

int arr[]=new int[5];

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

System.***out***.println("Enter your content: ");

arr[i] = sc.nextInt();

}

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

System.***out***.print(arr[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.

Ans= package array\_assign;

import java.util.Scanner;

public class Q2 {

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

public void acceptRecord(int arr[]) {

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

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

}

}

public void printRecord(int arr[]) {

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

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

}

}

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Q2 q = new Q2();

int arr[] = new int[5];

q.acceptRecord(arr);

q.printRecord(arr);

}

}

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

Ans = package array\_assign;

public class Q3 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int maxx = Integer.***MAX\_VALUE***;

int arr[] = {45,98,56,23,11,100,45,3};

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

if(arr[i]<maxx) {

maxx = arr[i];

}

}

System.***out***.println("Min number is : "+ maxx);

int minn = Integer.***MIN\_VALUE***;

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

if(arr[i]>minn) {

minn = arr[i];

}

}

System.***out***.println("Max number is : "+ minn);

}

}

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

package array\_assign;

import java.util.Arrays;

public class Q4 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int arr[] = {2,3,4,5,6,2,3,4,8,9,1};

Arrays.*sort*(arr);

int j =0;

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

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

j++;

arr[j]=arr[i];

}

}

int count = j+1;

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

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

}

}

}

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

ans= package array\_assign;

import java.util.ArrayList;

import java.util.Scanner;

public class Q6 {

public static void main(String[] args) {

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

int[] arr1 = new int[4];

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

for (int i = 0; i < arr1.length; i++) {

arr1[i] = sc.nextInt();

}

int[] arr2 = new int[5];

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

for (int i = 0; i < arr2.length; i++) {

arr2[i] = sc.nextInt();

}

ArrayList<Integer> intersection = new ArrayList<>();

for (int i = 0; i < arr1.length; i++) {

for (int j = 0; j < arr2.length; j++) {

if (arr1[i] == arr2[j] && !intersection.contains(arr1[i])) {

intersection.add(arr1[i]);

break;

}

}

}

// Output the intersection result

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

for (int num : intersection) {

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

}

sc.close();

}

}

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

Ans= public class Q5 {

public static int calculate(int n , int arr[]) {

int count =0;

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

count+=arr[i];

}

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

int result = AP - count;

return result;

}

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int arr[] = {1,2,3,4,5,6,7,9};

System.***out***.println(*calculate*(9, arr));

}

}

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.

Ans: package array\_assign;

import java.util.Scanner;

class Test{

int max;

int arr[]= new int[4];

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

public Test() {

System.***out***.println("Enter the elements: ");

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

arr[i] = sc.nextInt();

}

}

public int acceptRecord() {

this.max = arr[0];

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

if (arr[i]>max) {

max=arr[i];

}

}

return max;

}

public void printRecord() {

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

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

}

System.***out***.println("Max element: "+ this.max);

}

}

public class Q7 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Test t = new Test();

t.acceptRecord();

t.printRecord();

}

}

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

Ans : import java.util.Scanner;

class Test{

int arr[]= new int[4];

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

public Test() {

}

public int[] getArr() {

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

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

}

return arr;

}

public void setArr(int[] arr) {

System.***out***.println("Enter the elements: ");

this.arr[0]=sc.nextInt();

this.arr[1]=sc.nextInt();

this.arr[2]=sc.nextInt();

this.arr[3]=sc.nextInt();

}

}

public class Q7 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Test t = new Test();

t.setArr(null);

t.getArr();

}

}

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.

Ans: import java.util.Scanner;

public class AirplaneSeatingSystem {

private char[][] seats;

private int rows, cols;

public AirplaneSeatingSystem(int rows, int cols) {

this.rows = rows;

this.cols = cols;

seats = new char[rows][cols];

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

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

seats[i][j] = 'A'; // 'A' stands for available

}

}

}

public void bookSeat(int row, int col) {

if (isSeatValid(row, col)) {

if (seats[row][col] == 'A') {

seats[row][col] = 'O'; // 'O' stands for occupied

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

} else {

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

}

}

}

public void cancelBooking(int row, int col) {

if (isSeatValid(row, col)) {

if (seats[row][col] == 'O') {

seats[row][col] = 'A'; // Mark the seat as available again

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

} else {

System.out.println("Seat (" + row + ", " + col + ") is not occupied.");

}

}

}

public void checkSeatAvailability(int row, int col) {

if (isSeatValid(row, col)) {

if (seats[row][col] == 'A') {

System.out.println("Seat (" + row + ", " + col + ") is available.");

} else {

System.out.println("Seat (" + row + ", " + col + ") is occupied.");

}

}

}

public void displaySeatingChart() {

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

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

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

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

}

System.out.println(); // Move to the next row

}

}

private boolean isSeatValid(int row, int col) {

if (row < 0 || row >= rows || col < 0 || col >= cols) {

System.out.println("Invalid seat coordinates.");

return false;

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int rows = sc.nextInt();

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

int cols = sc.nextInt();

AirplaneSeatingSystem system = new AirplaneSeatingSystem(rows, cols);

boolean exit = false;

while (!exit) {

System.out.println("\nMenu:");

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

System.out.println("2. Cancel 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 = sc.nextInt();

switch (choice) {

case 1:

// Book a seat

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

int bookRow = sc.nextInt();

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

int bookCol = sc.nextInt();

system.bookSeat(bookRow, bookCol);

break;

case 2:

// Cancel a booking

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

int cancelRow = sc.nextInt();

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

int cancelCol = sc.nextInt();

system.cancelBooking(cancelRow, cancelCol);

break;

case 3:

// Check seat availability

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

int checkRow = sc.nextInt();

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

int checkCol = sc.nextInt();

system.checkSeatAvailability(checkRow, checkCol);

break;

case 4:

// Display seating chart

system.displaySeatingChart();

break;

case 5:

// Exit the program

exit = true;

System.out.println("Exiting the program.");

break;

default:

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

}

}

sc.close();

}

}