**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.

Code :-

**package** project;

**import** java.util.Scanner;

**public** **class** Program1 {

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

// Declare a single-dimensional array of 5 integers

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

// Print the default values of the array

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

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

System.***out***.println("Index " + i + ": " + numbers[i]);

}

// Create a Scanner

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

// Accept records from the user

System.***out***.println("Enter 5 integers to update the array:");

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

System.***out***.print("Enter value for index " + i + ": ");

numbers[i] = scanner.nextInt();

}

// Print the updated values of the array

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

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

System.***out***.println("Index " + i + ": " + numbers[i]);

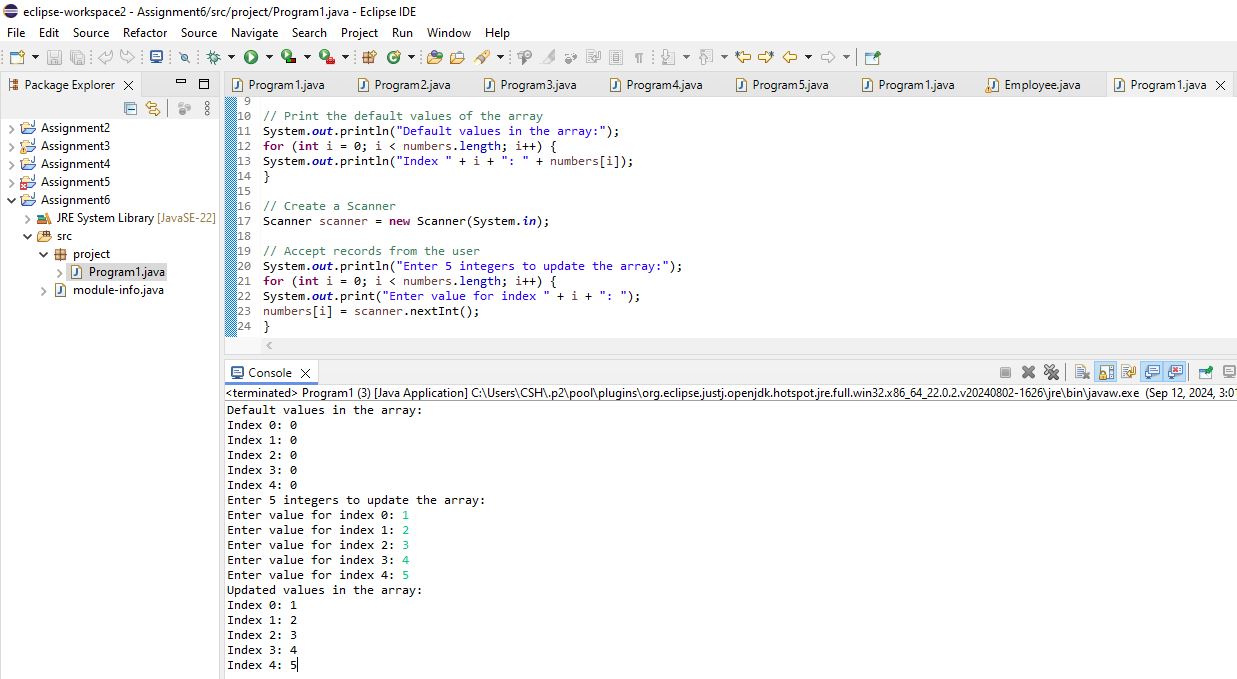
}

scanner.close();

}

}

Output –



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** project;

**import** java.util.Scanner;

**public** **class** Program2 {

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

// Declare a single-dimensional array of 5 integers

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

// Call method to accept input

*acceptRecord*(numbers);

// Call method to print the state of array

*printRecord*(numbers);

}

// Method to accept input from the user and update the array

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

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

System.***out***.println("Enter 5 integers to update the array:");

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

System.***out***.print("Enter value for index " + i + ": ");

array[i] = scanner.nextInt();

}

}

// Method to print the state of the array

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

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

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

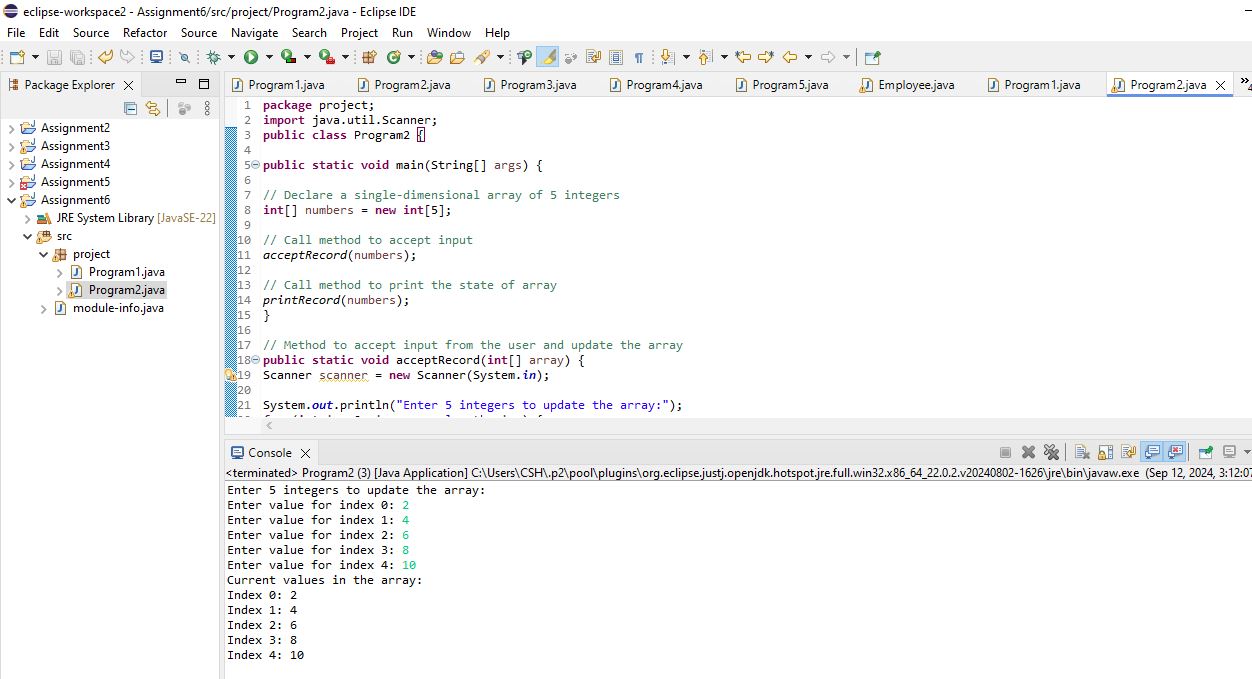
System.***out***.println("Index " + i + ": " + array[i]);

}

}

}

Output –



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

Code :-

**package** project;

**public** **class** Program3 {

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

**int**[] numbers = {40, 23, 12, 92, 45, 67, 23};

**int** max = *findMax*(numbers);

**int** min = *findMin*(numbers);

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]; // Initialize min to the first element

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

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

min = array[i];

}

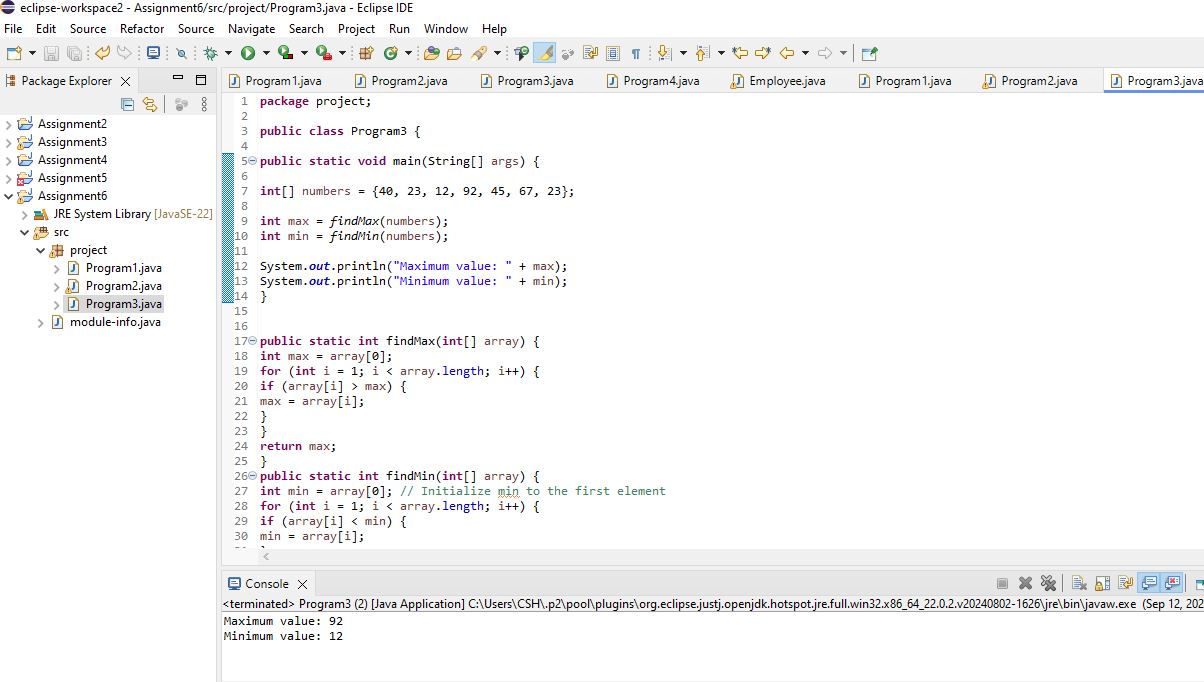
}

**return** min;

}

}

Output –



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

Code :-

package project;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

public class Program4 {

public static void main(String[] args) {

{

int[] numbers = {10, 20, 10, 30, 40, 30, 50};

int[] uniqueNumbers = removeDuplicates(numbers);

System.out.println("Array with duplicates removed: " + Arrays.toString(uniqueNumbers));

}

}

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

Set<Integer> uniqueSet = new HashSet<>();

for (int number : array) {

uniqueSet.add(number);

}

int[] resultArray = new int[uniqueSet.size()];

int index = 0;

for (int number : uniqueSet) {

resultArray[index++] = number;

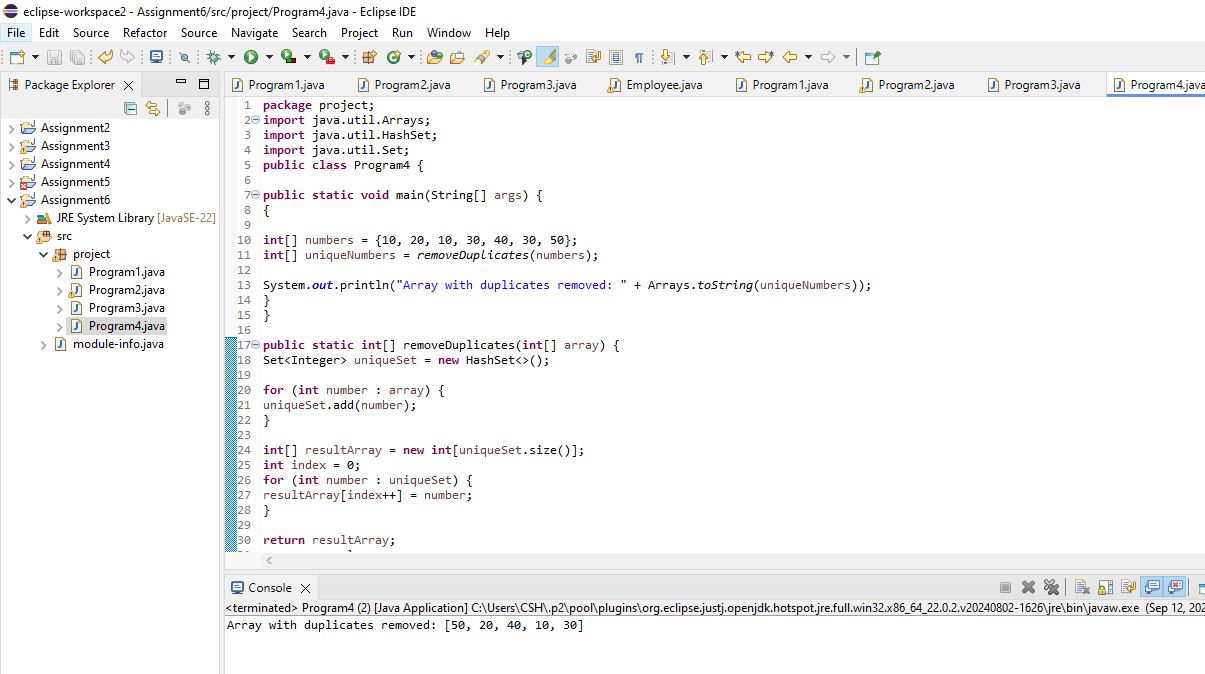
}

return resultArray;

}

}

Output –



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

Code :-

package project;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

public class Program5 {

public static void main(String[] args) {

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

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

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

System.out.println("Intersection of the two arrays: " + Arrays.toString(intersection));

}

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

Set<Integer> set1 = new HashSet<>();

for (int num : array1) {

set1.add(num);

}

Set<Integer> intersectionSet = new HashSet<>();

for (int num : array2) {

if (set1.contains(num)) {

intersectionSet.add(num);

}

}

int[] resultArray = new int[intersectionSet.size()];

int index = 0;

for (int num : intersectionSet) {

resultArray[index++] = num;

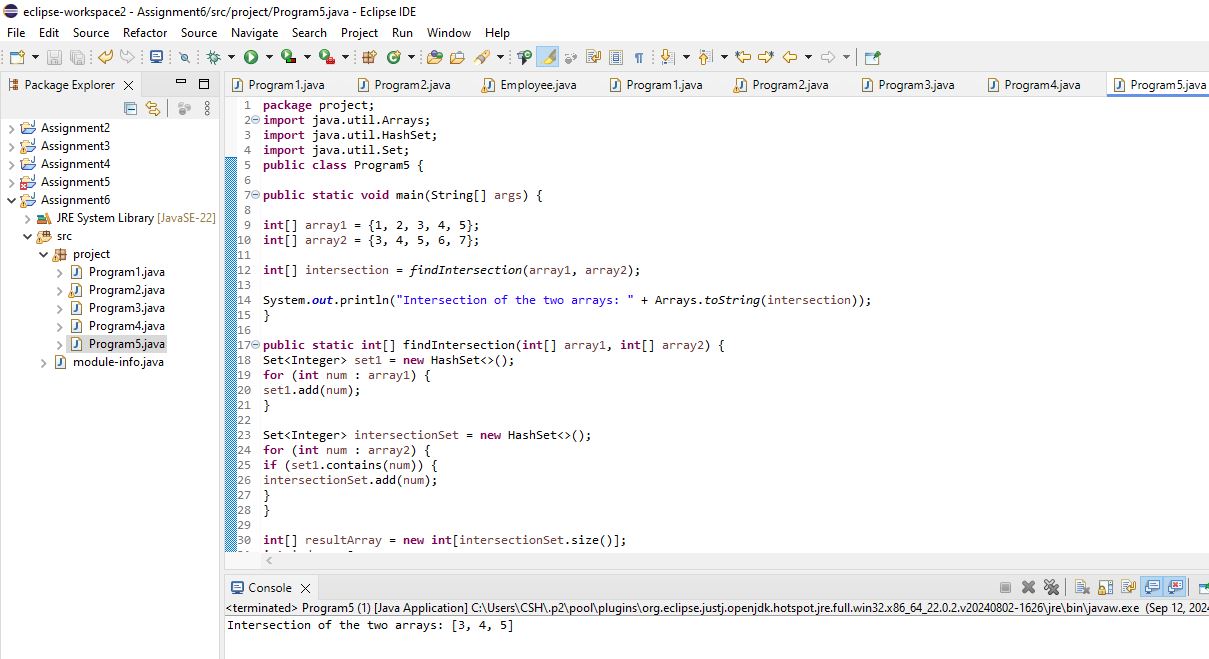
}

return resultArray;

}

}

Output –



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

Code :-

**package** project;

**public** **class** Program6 {

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

**int**[] numbers = {1, 2, 4, 5, 6};

**int** N = 6;

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

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

}

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

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

**int** actualSum = 0;

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

actualSum += num;

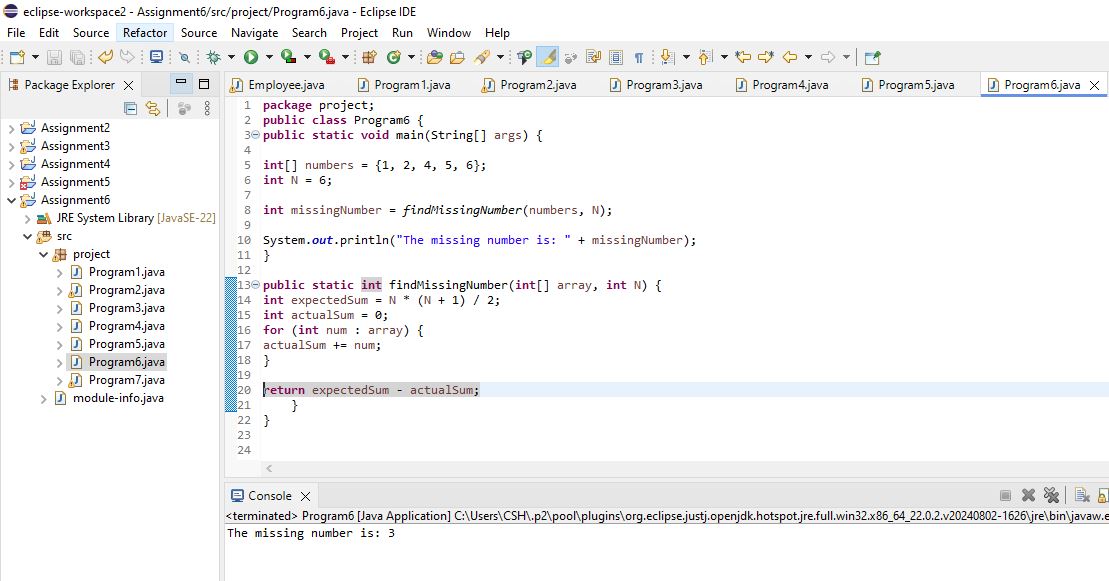
}

**return** expectedSum - actualSum;

}

}

Output –



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** project;

**import** java.util.Scanner;

**public** **class** Program7 {

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

**public** Program7(**int** size) {

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

}

**public** **void** acceptRecord() {

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

System.***out***.println("Enter " + numbers.length + " integers to update the array:");

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

System.***out***.print("Enter value for index " + i + ": ");

numbers[i] = scanner.nextInt();

}

}

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

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

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

System.***out***.println("Index " + i + ": " + numbers[i]);

}

}

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

Program7 manager = **new** Program7(5);

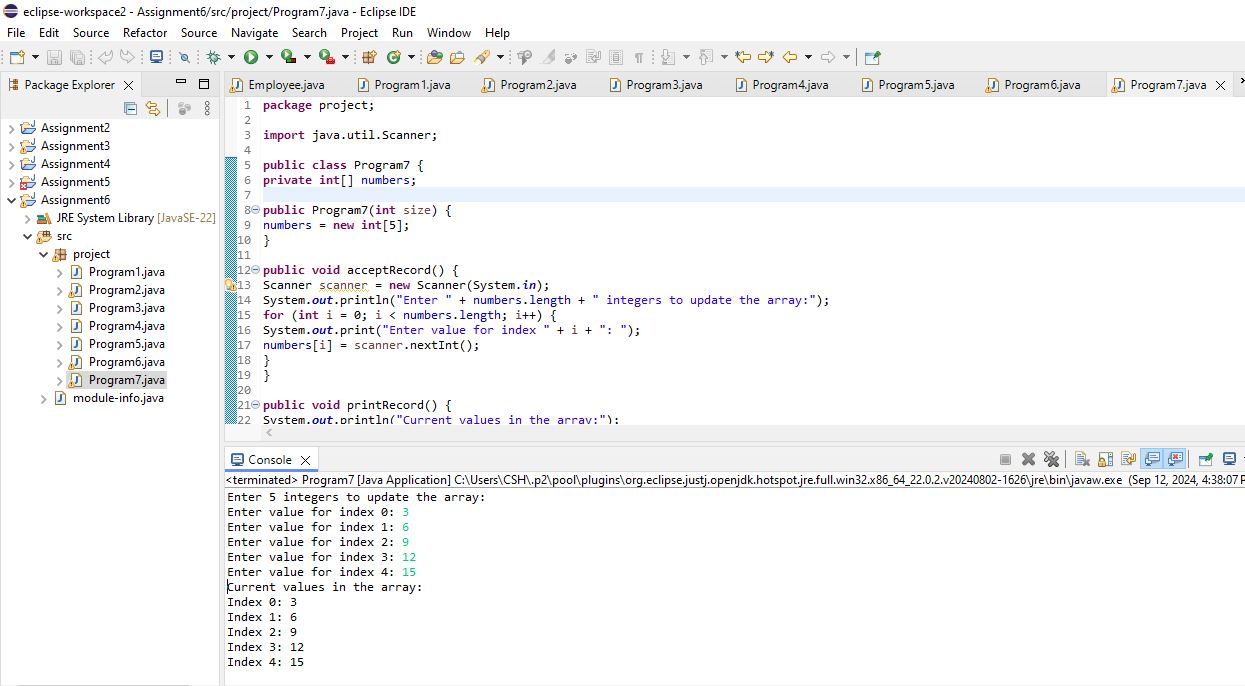
manager.acceptRecord();

manager.printRecord();

}

}

Output –



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

Code:-

**package** project;

**import** java.util.Scanner;

**public** **class** Program8 {

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

**public** Program8(**int** size) {

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

}

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

**return** numbers;

}

**public** **void** setNumbers() {

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

System.***out***.println("Enter " + numbers.length + " integers to update the array:");

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

System.***out***.print("Enter value for index " + i + ": ");

numbers[i] = scanner.nextInt();

}

}

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

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

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

System.***out***.println("Index " + i + ": " + numbers[i]);

}

}

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

Program8 manager = **new** Program8(5);

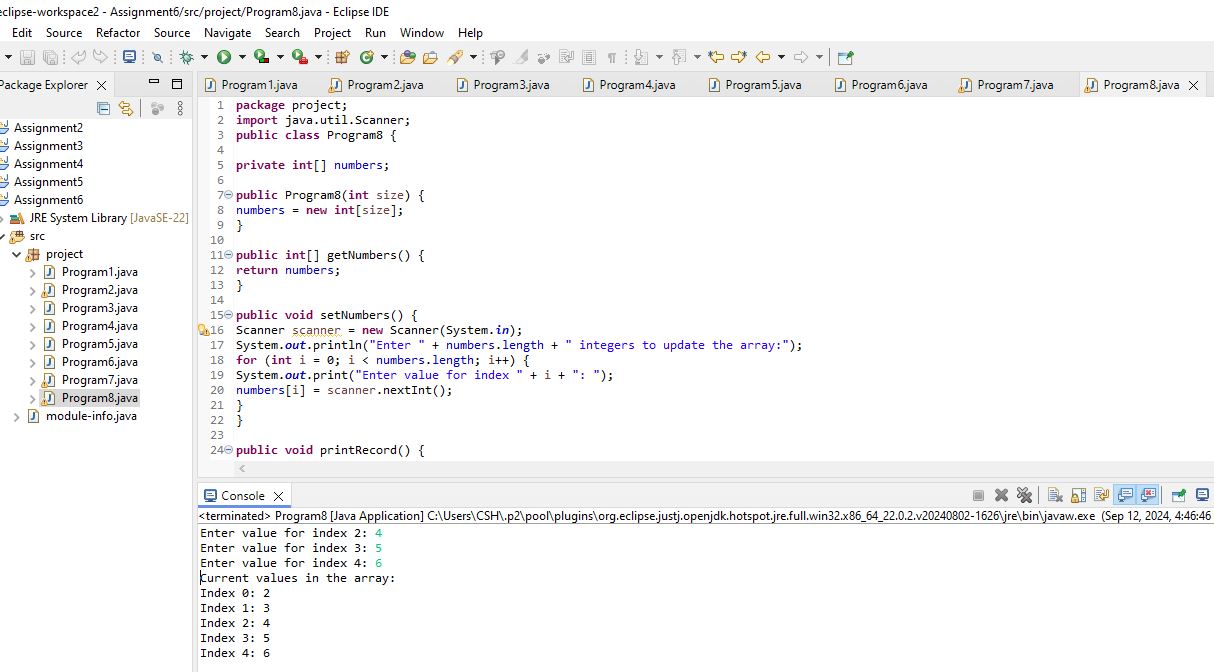
manager.setNumbers();

manager.printRecord();

}

}

Output –



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** project;

**import** java.util.Scanner;

**public** **class** Program9 {

**private** String[][] seats;

**public** Program9(**int** rows, **int** columns) {

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

initializeSeats();

}

**private** **void** initializeSeats() {

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

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

seats[i][j] = "Available";

}

}

}

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

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

**if** (seats[row][column].equals("Available")) {

seats[row][column] = "Booked";

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

}

**else** {

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

}

}

}

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

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

**if** (seats[row][column].equals("Booked")) {

seats[row][column] = "Available";

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

} **else** {

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

}

}

}

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

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

**return** seats[row][column].equals("Available");

}

**return** **false**;

}

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

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

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

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

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

}

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

}

}

**private** **boolean** isSeatValid(**int** row, **int** column) {

**if** (row >= 0 && row < seats.length && column >= 0 && column < seats[0].length) {

**return** **true**;

}

**else** {

System.***out***.println("Invalid seat position: (" + row + ", " + column + ")");

**return** **false**;

}

}

**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();

Program9 program = **new** Program9(rows, columns);

// Test the functionalities

program.displaySeatingChart();

// Book a seat

System.***out***.print("Enter row and column to book a seat: ");

**int** rowToBook = scanner.nextInt();

**int** colToBook = scanner.nextInt();

program.bookSeat(rowToBook, colToBook);

program.displaySeatingChart();

// Cancel a booking

System.***out***.print("Enter row and column to cancel a seat booking: ");

**int** rowToCancel = scanner.nextInt();

**int** colToCancel = scanner.nextInt();

program.cancelBooking(rowToCancel, colToCancel);

program.displaySeatingChart();

// Check seat availability

System.***out***.print("Enter row and column to check seat availability: ");

**int** rowToCheck = scanner.nextInt();

**int** colToCheck = scanner.nextInt();

**boolean** available = program.isSeatAvailable(rowToCheck, colToCheck);

System.***out***.println("Seat (" + rowToCheck + ", " + colToCheck + ") is " + (available ? "Available" : "Booked"));

scanner.close();

}

}

Output –

