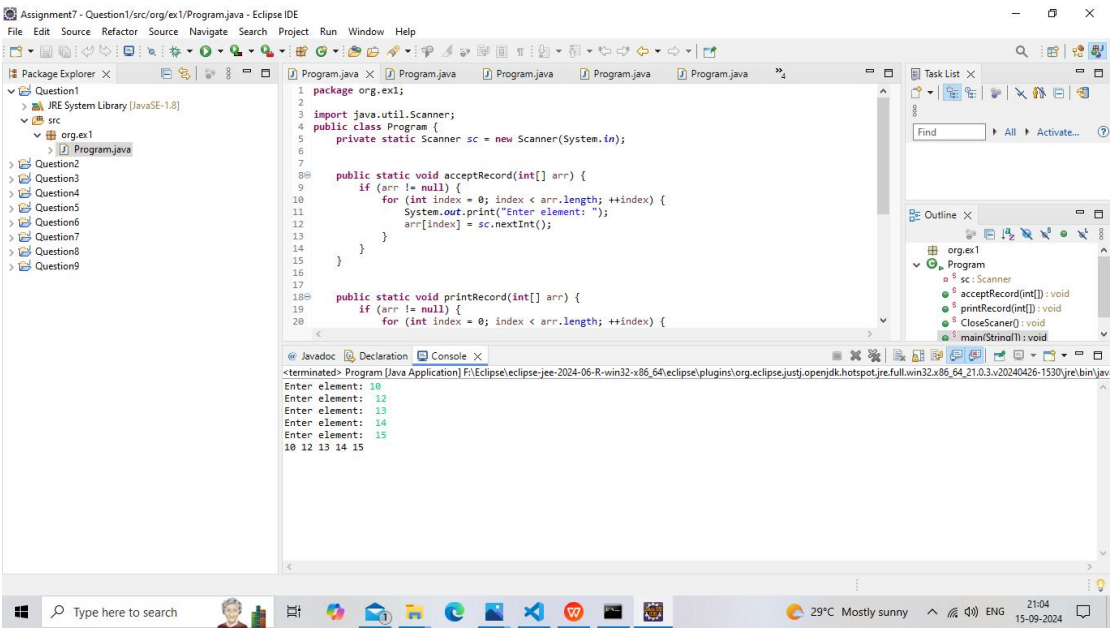
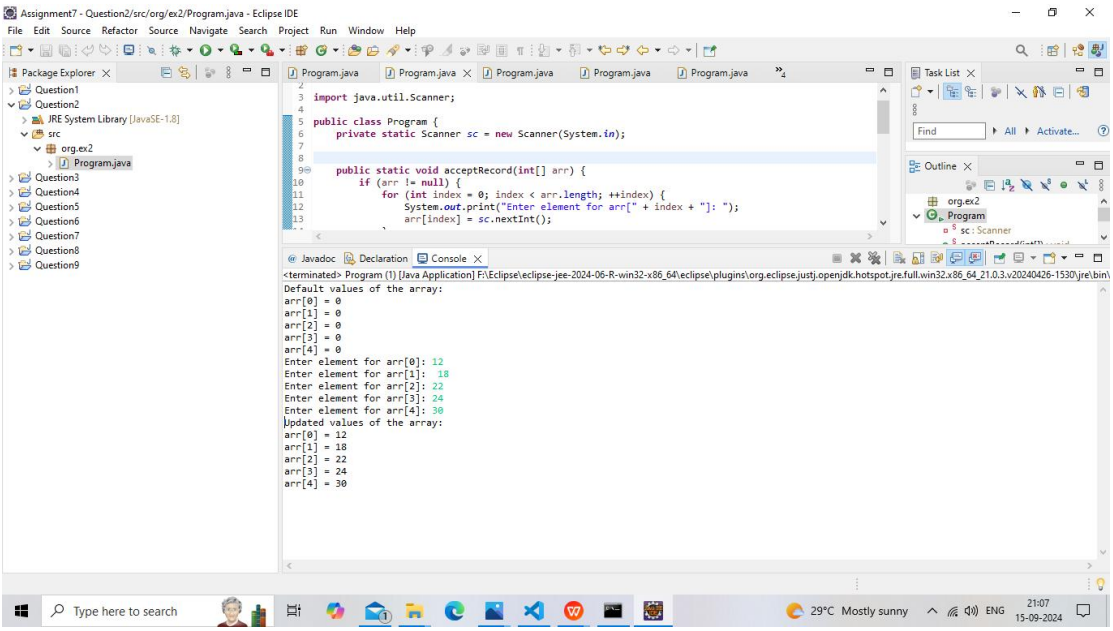
Question1

```
package org.ex1;

import java.util.Scanner;
public class Program {
private static Scanner sc = new Scanner(System.in);

public static void acceptRecord(int[] arr) {
if (arr != null) {
for (int index = 0; index < arr.length; ++index) {
System.out.print("Enter element: ");
arr[index] = sc.nextInt();
}
}
}

public static void printRecord(int[] arr) {
if (arr != null) {
for (int index = 0; index < arr.length; ++index) {
System.out.print(arr[index] + " ");
}
System.out.println();
}
}
public static void CloseScaner() {
sc.close();
}

public static void main(String[] args) {
int[] arr = new int[5];
Program.acceptRecord(arr);
Program.printRecord(arr);
Program.CloseScaner();
}
}
```



Question2

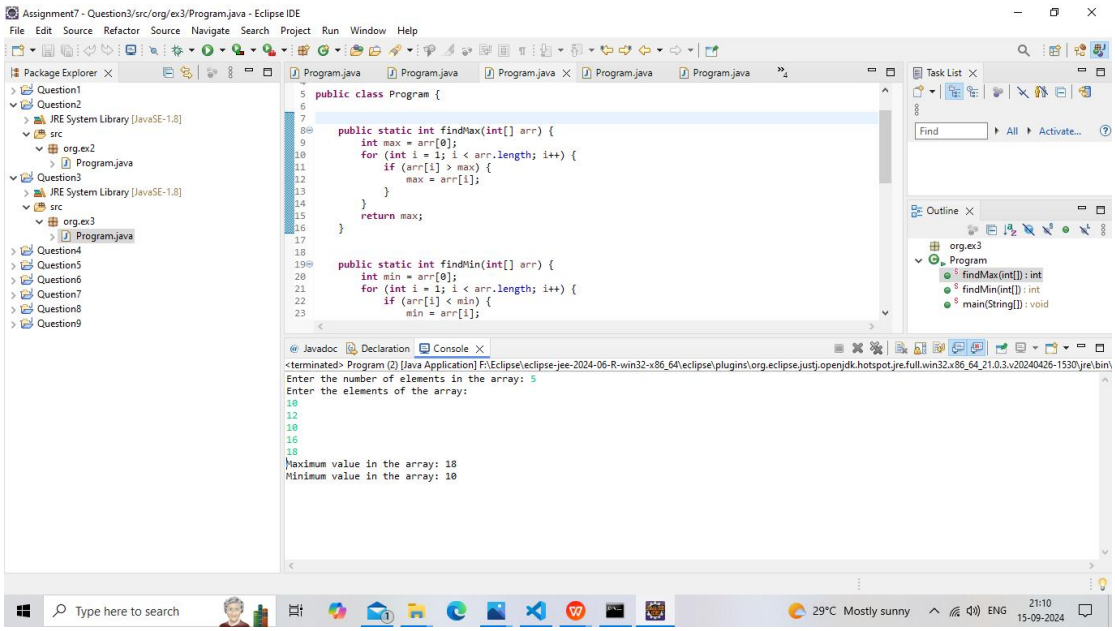
```
package org.ex2;

import java.util.Scanner;

public class Program {
private static Scanner sc = new Scanner(System.in);

public static void acceptRecord(int[] arr) {
if (arr != null) {
for (int index = 0; index < arr.length; ++index) {
System.out.print("Enter element for arr[" + index + "]: ");
arr[index] = sc.nextInt();
}
}
}

public static void printRecord(int[] arr) {
if (arr != null) {
for (int index = 0; index < arr.length; ++index) {
System.out.println("arr[" + index + "] = " + arr[index]);
}
}
}

public static void main(String[] args) {
int[] arr = new int[5];

System.out.println("Default values of the array:");
Program.printRecord(arr);

Program.acceptRecord(arr);

System.out.println("Updated values of the array:");
Program.printRecord(arr);
}
}
```



Question3

```
package org.ex3;

import java.util.Scanner;

public class Program {

public static int findMax(int[] arr) {
int max = arr[0];
for (int i = 1; i < arr.length; i++) {
if (arr[i] > max) {
max = arr[i];
}
}
return max;
}

public static int findMin(int[] arr) {
int min = arr[0];
for (int i = 1; i < arr.length; i++) {
if (arr[i] < min) {
min = arr[i];
}
}
return min;
}

public static void main(String[] args) {
Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of elements in the array: ");
int n = sc.nextInt();

int[] arr = new int[n];
```

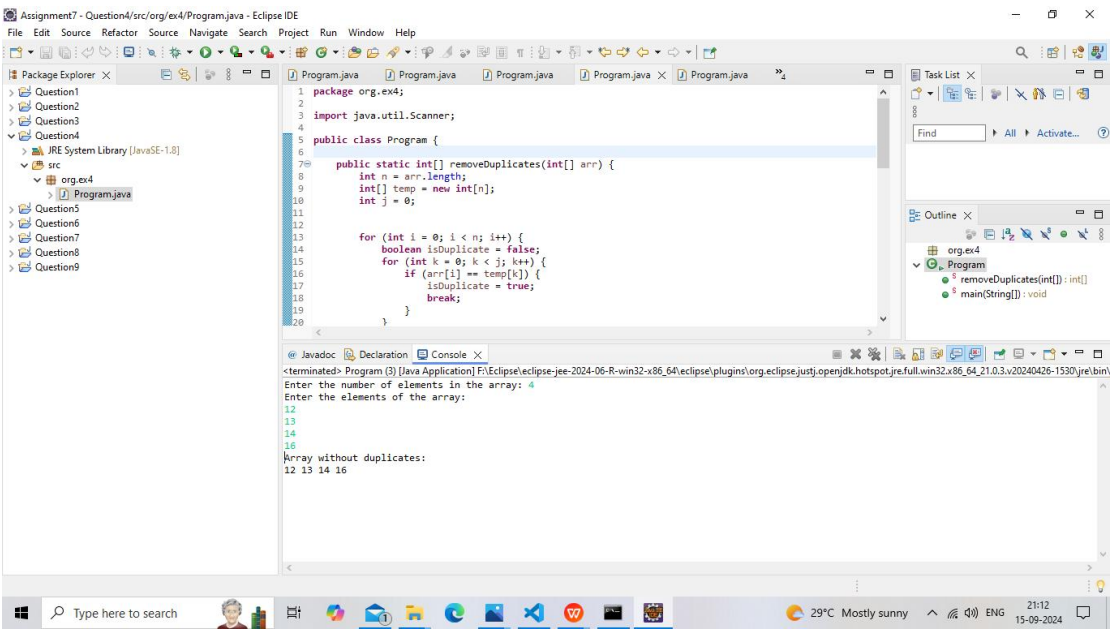
```
System.out.println("Enter the elements of the array:");
for (int i = 0; i < n; i++) {
arr[i] = sc.nextInt();
}

int maxValue = findMax(arr);
int minValue = findMin(arr);

System.out.println("Maximum value in the array: " + maxValue);
System.out.println("Minimum value in the array: " + minValue);

sc.close();
}
}
```



Question4

```
package org.ex4;

import java.util.Scanner;

public class Program {
public static int[] removeDuplicates(int[] arr) {
int n = arr.length;
int[] temp = new int[n];
int j = 0;

for (int i = 0; i < n; i++) {
boolean isDuplicate = false;
for (int k = 0; k < j; k++) {
if (arr[i] == temp[k]) {
isDuplicate = true;
break;
}
```

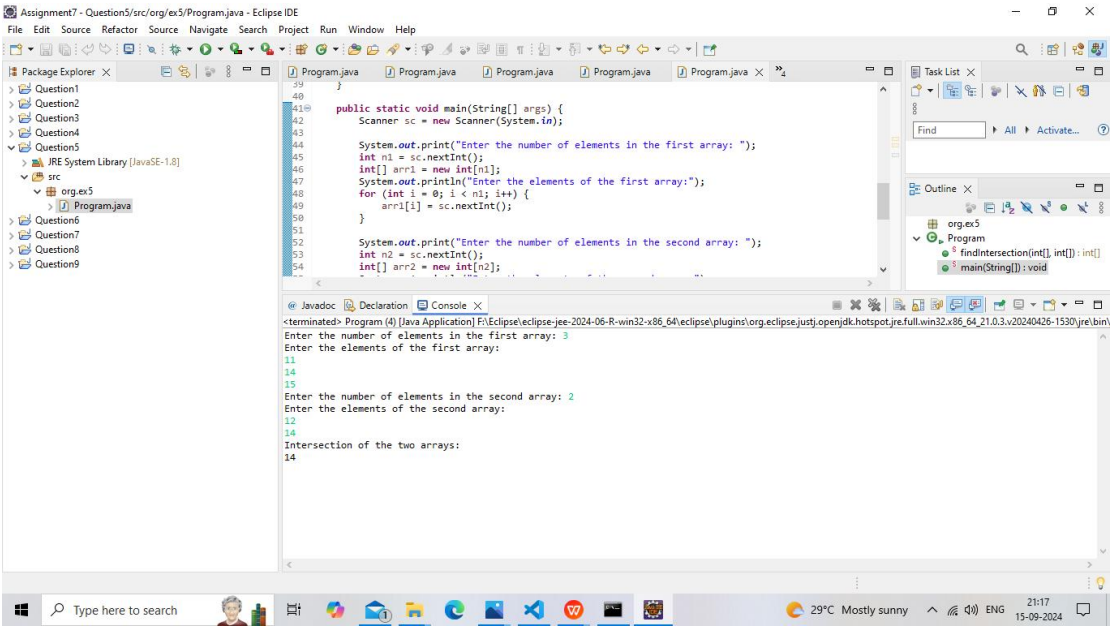
```
}
if (!isDuplicate) {
temp[j++] = arr[i];
}
}

int[] uniqueArray = new int[j];
for (int i = 0; i < j; i++) {
uniqueArray[i] = temp[i];
}
return uniqueArray;
}

public static void main(String[] args) {
Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of elements in the array: ");
int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter the elements of the array:");
for (int i = 0; i < n; i++) {
arr[i] = sc.nextInt();
}

int[] uniqueArray = removeDuplicates(arr);

System.out.println("Array without duplicates:");
for (int i = 0; i < uniqueArray.length; i++) {
System.out.print(uniqueArray[i] + " ");
}

sc.close();
}
}
```



Question5

```
package org.ex5;

import java.util.Scanner;

public class Program {
public static int[] findIntersection(int[] arr1, int[] arr2) {
int n1 = arr1.length;
int n2 = arr2.length;
int[] temp = new int[Math.min(n1, n2)];
int k = 0;

for (int i = 0; i < n1; i++) {
for (int j = 0; j < n2; j++) {
if (arr1[i] == arr2[j]) {
boolean alreadyExists = false;
for (int m = 0; m < k; m++) {
if (temp[m] == arr1[i]) {
alreadyExists = true;
break;
}
}
if (!alreadyExists) {
temp[k++] = arr1[i];
}
}
}
}
int[] intersectionArray = new int[k];
for (int i = 0; i < k; i++) {
intersectionArray[i] = temp[i];
}
return intersectionArray;
}

public static void main(String[] args) {
Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of elements in the first array: ");
int n1 = sc.nextInt();
int[] arr1 = new int[n1];
System.out.println("Enter the elements of the first array:");
for (int i = 0; i < n1; i++) {
arr1[i] = sc.nextInt();
}

System.out.print("Enter the number of elements in the second array: ");
int n2 = sc.nextInt();
int[] arr2 = new int[n2];
System.out.println("Enter the elements of the second array:");
for (int i = 0; i < n2; i++) {
arr2[i] = sc.nextInt();
}

int[] intersection = findIntersection(arr1, arr2);
```

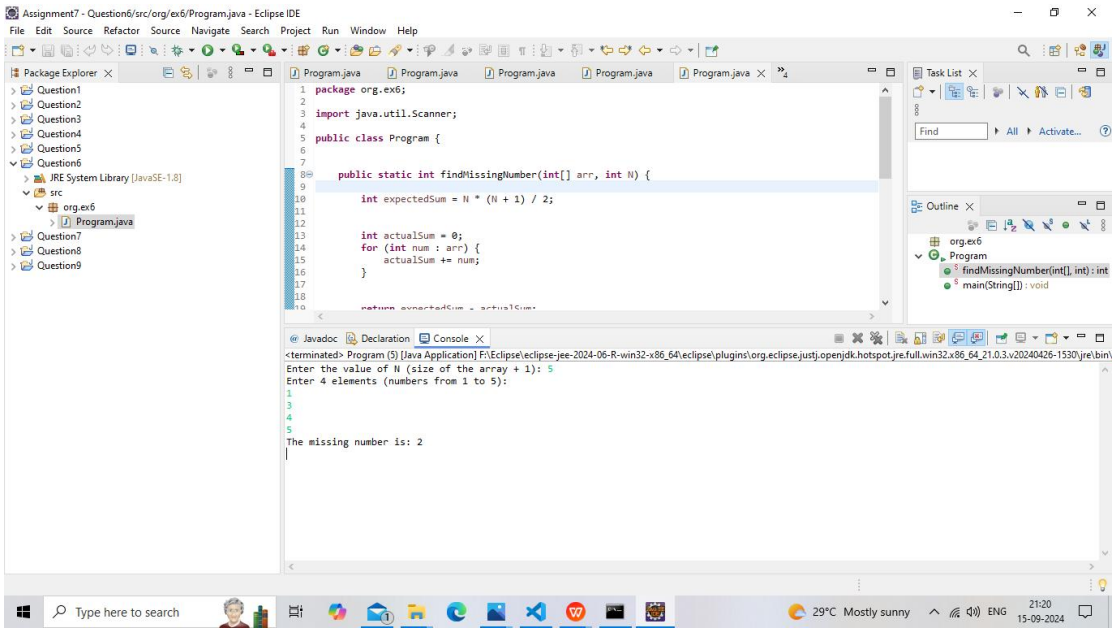
```
System.out.println("Intersection of the two arrays:");
for (int i = 0; i < intersection.length; i++) {
System.out.print(intersection[i] + " ");
}

sc.close();
}
}
```



Question6

```
package org.ex6;

import java.util.Scanner;

public class Program {

public static int findMissingNumber(int[] arr, int N) {
int expectedSum = N * (N + 1) / 2;
int actualSum = 0;
for (int num : arr) {
actualSum += num;
}
return expectedSum - actualSum;
}

public static void main(String[] args) {
Scanner sc = new Scanner(System.in);

System.out.print("Enter the value of N (size of the array + 1): ");
int N = sc.nextInt();
int[] arr = new int[N - 1];
System.out.println("Enter " + (N - 1) + " elements (numbers from 1 to " + N
+ "):");
```

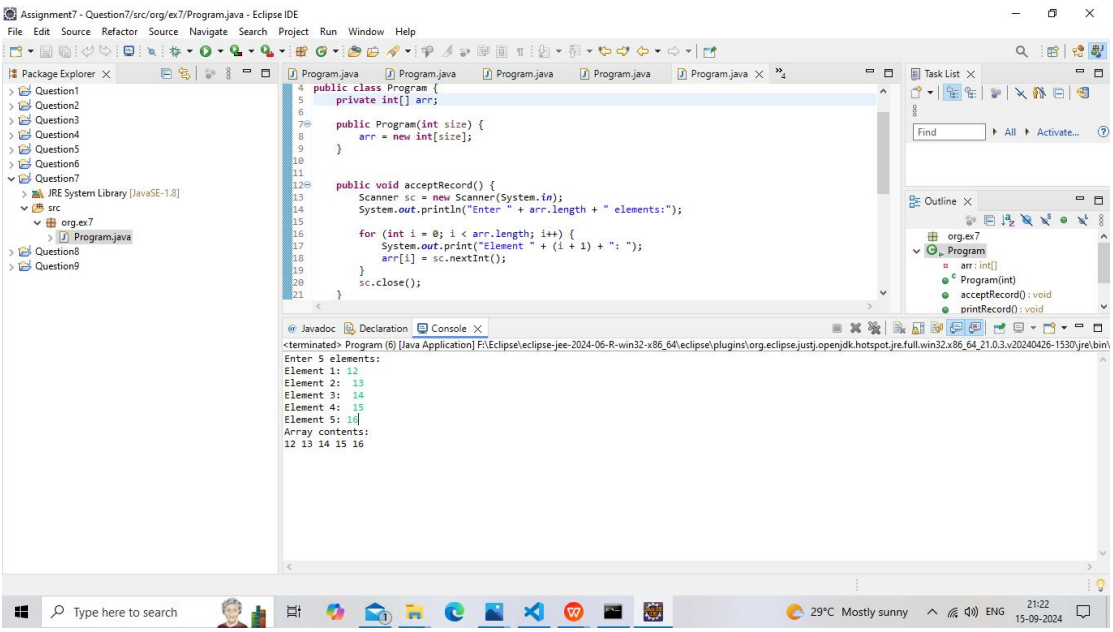
```
for (int i = 0; i < N - 1; i++) {
arr[i] = sc.nextInt();
}
int missingNumber = findMissingNumber(arr, N);
System.out.println("The missing number is: " + missingNumber);

sc.close();
}
}
```



Question7

```
package org.ex1;

import java.util.Scanner;
public class Program {
private static Scanner sc = new Scanner(System.in);

public static void acceptRecord(int[] arr) {
if (arr != null) {
for (int index = 0; index < arr.length; ++index) {
System.out.print("Enter element: ");
arr[index] = sc.nextInt();
}
}
}

public static void printRecord(int[] arr) {
if (arr != null) {
for (int index = 0; index < arr.length; ++index) {
System.out.print(arr[index] + " ");
}
System.out.println();
}
}
```

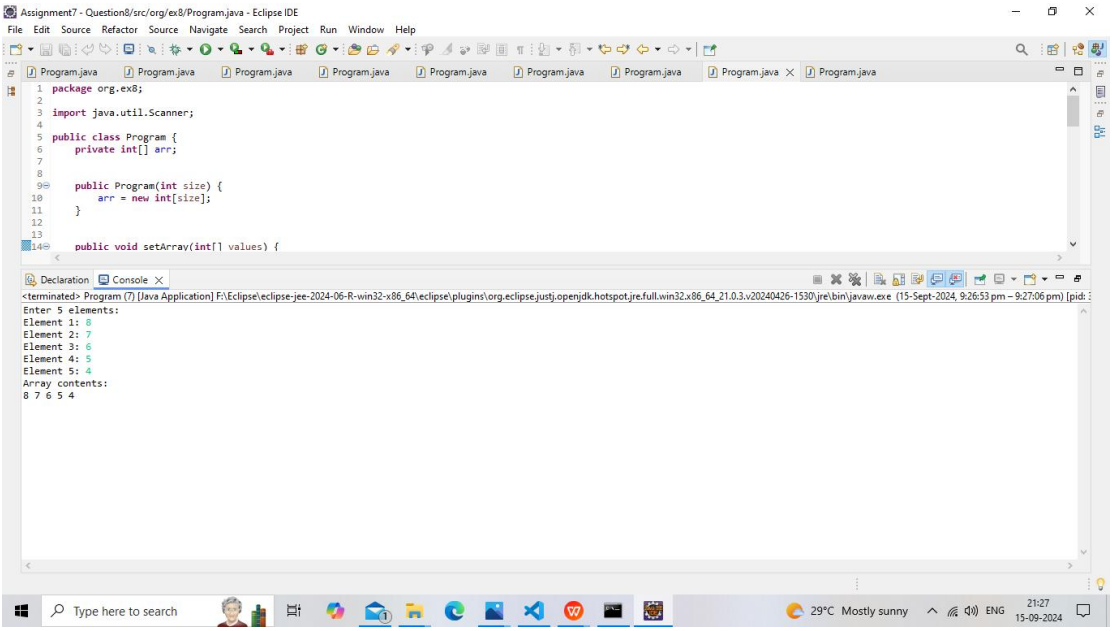
```
public static void CloseScaner() {
sc.close();
}

public static void main(String[] args) {
int[] arr = new int[5];
Program.acceptRecord(arr);
Program.printRecord(arr);
Program.CloseScaner();
}
}
```



Question8

```
package org.ex8;

import java.util.Scanner;

public class Program {
private int[] arr;

public Program(int size) {
arr = new int[size];
}

public void setArray(int[] values) {
if (values.length == arr.length) {
for (int i = 0; i < arr.length; i++) {
arr[i] = values[i];
}
} else {
System.out.println("Error: Array size mismatch.");
}
}
```

```
public int[] getArray() {
return arr;
}

public void printArray() {
System.out.println("Array contents:");
for (int value : arr) {
System.out.print(value + " ");
}
System.out.println();
}

public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
Program program = new Program(5);

System.out.println("Enter " + program.getArray().length + " elements:");
int[] inputArray = new int[program.getArray().length];
for (int i = 0; i < inputArray.length; i++) {
System.out.print("Element " + (i + 1) + ": ");
inputArray[i] = sc.nextInt();
}

program.setArray(inputArray);

program.printArray();
sc.close();
}
}
```

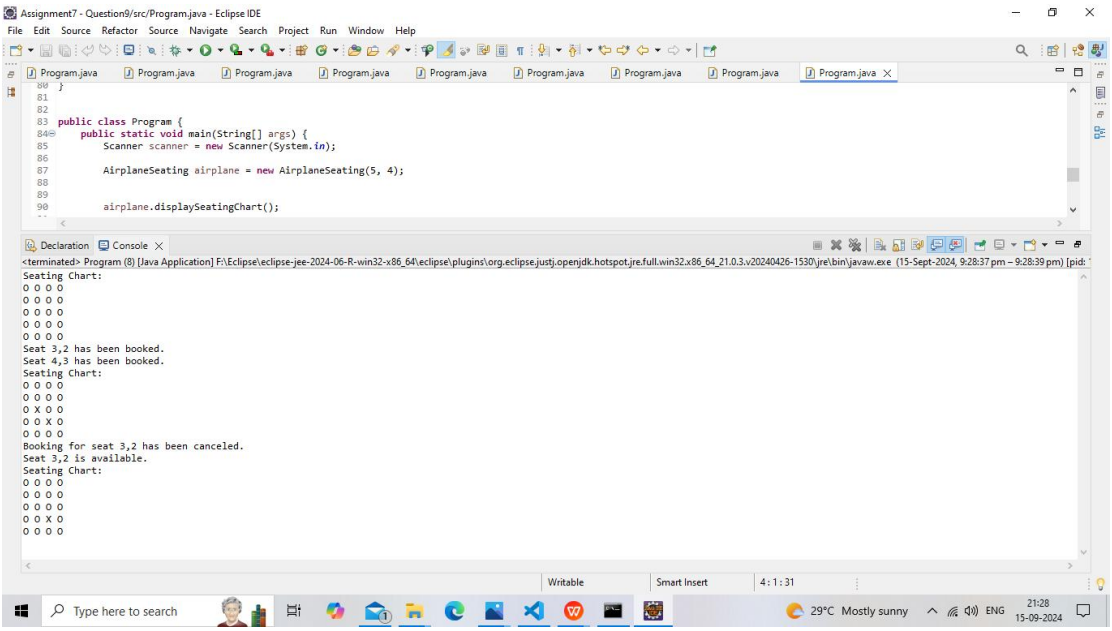


Question9

```
import java.util.Scanner;
```

```
class AirplaneSeating {
private char[][] seats;
private int rows;
private int columns;

public AirplaneSeating(int rows, int columns) {
this.rows = rows;
this.columns = columns;
seats = new char[rows][columns];

for (int i = 0; i < rows; i++) {
for (int j = 0; j < columns; j++) {
seats[i][j] = 'O';
}
}
}

public boolean bookSeat(int row, int col) {
if (isSeatValid(row, col)) {
if (seats[row][col] == 'O') {
seats[row][col] = 'X';
System.out.println("Seat " + (row + 1) + "," + (col + 1) + " has been
booked.");
return true;
} else {
System.out.println("Seat " + (row + 1) + "," + (col + 1) + " is already
booked.");
}
} else {
System.out.println("Invalid seat selection.");
}
return false;
}

public boolean cancelSeat(int row, int col) {
if (isSeatValid(row, col)) {
if (seats[row][col] == 'X') {
seats[row][col] = 'O';
System.out.println("Booking for seat " + (row + 1) + "," + (col + 1) + "
has been canceled.");
return true;
} else {
System.out.println("Seat " + (row + 1) + "," + (col + 1) + " is not
booked.");
}
} else {
System.out.println("Invalid seat selection.");
}
return false;
}


public boolean isSeatAvailable(int row, int col) {
if (isSeatValid(row, col)) {
return seats[row][col] == 'O';
}
```

```
System.out.println("Invalid seat selection.");
return false;
}

public void displaySeatingChart() {
System.out.println("Seating Chart:");
for (int i = 0; i < rows; i++) {
for (int j = 0; j < columns; j++) {
System.out.print(seats[i][j] + " ");
}
System.out.println();
}
}

private boolean isSeatValid(int row, int col) {
return row >= 0 && row < rows && col >= 0 && col < columns;
}
}



public class Program {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
AirplaneSeating airplane = new AirplaneSeating(5, 4);

airplane.displaySeatingChart();



airplane.bookSeat(2, 1);
airplane.bookSeat(3, 2);

airplane.displaySeatingChart();

airplane.cancelSeat(2, 1);

if (airplane.isSeatAvailable(2, 1)) {
System.out.println("Seat 3,2 is available.");
} else {
System.out.println("Seat 3,2 is not available.");
}

airplane.displaySeatingChart();

scanner.close();
}
}
```


Assignment7 - Question9/src/Program.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Program.java
}
public class Program {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
AirplaneSeating airplane = new AirplaneSeating(5, 4);
airplane.displaySeatingChart();
Declaration Console
<terminated> Program (8) [Java Application] F:\Eclipse\eclipse-jee-2024-06-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw.exe (15-Sept-2024, 9:28:37 pm – 9:28:39 pm) [pid:
Seating Chart:
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
Seat 3,2 has been booked.
Seat 4,3 has been booked.
Seating Chart:
0 0 0 0
0 0 0 0
0 X 0 0
0 0 X 0
0 0 0 0
Booking for seat 3,2 has been canceled.
Seat 3,2 is available.
Seating Chart:
0 0 0 0
0 0 0 0
0 0 0 0
0 0 X 0
0 0 0 0
Writable
Smart Insert
4 : 1 : 31
Type here to search
29°C Mostly sunny
ENG
21:28
15-09-2024