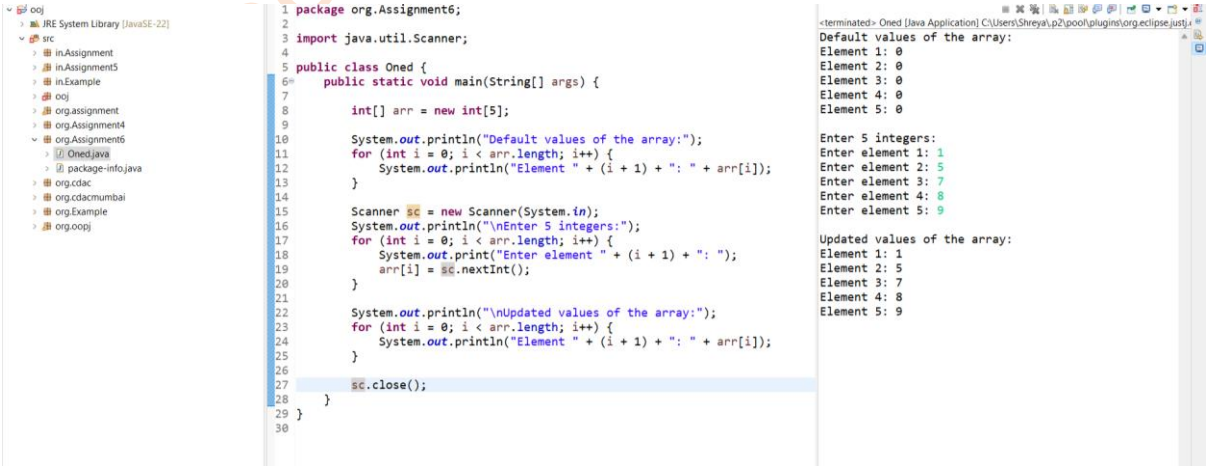
ASSIGNMENT NO.6

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.**

```
package org.Assignment6;
import java.util.Scanner;
public class Oned {
   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("Element " + (i + 1) + ": " + arr[i]);
      }

      Scanner sc = new Scanner(System.in);
      System.out.println("Enter 5 integers:");
      for (int i = 0; i < arr.length; i++) {
         System.out.print("Enter element " + (i + 1) + ": ");
         arr[i] = sc.nextInt();
      }

      System.out.println("Updated values of the array:");
      for (int i = 0; i < arr.length; i++) {
         System.out.println("Element " + (i + 1) + ": " + arr[i]);
      }

      sc.close();
   }
}
```

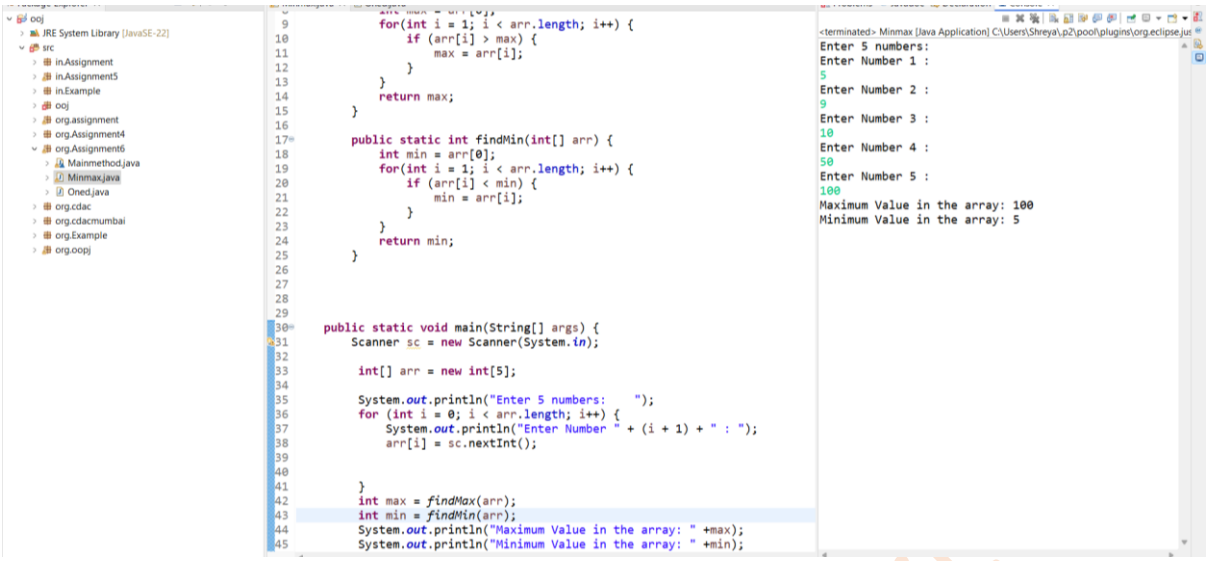


2. Declare a single-dimensional array of 5 integers inside the `main` method. Define a method named `acceptRecord` to get input from the terminal into the array and another method named `printRecord` to print the state of the array to the terminal.

```
package org.Assignment6;
import java.util.Scanner;
 class Mainmethod {

        public static void acceptRecord(int[] arr) {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter 5 Numbers: ");
            for(int i = 0; i < arr.length; i++) {
            System.out.println("Enter Number " +(i + 1) + ":");
            arr[i] = sc.nextInt();
            }
        }
        public static void printRecord(int[] arr) {
            Scanner sc = new Scanner(System.in);
            System.out.println("The Numbers in the Array are: ");
            for(int i = 0; i < arr.length; i++) {
                System.out.println("Number" + (i + 1) + ":" +arr[i]);
            }
        }
        public static void main(String[] args) {
            int[] arr = new int[5];
            acceptRecord(arr);
            printRecord(arr);
        }
    }
```

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

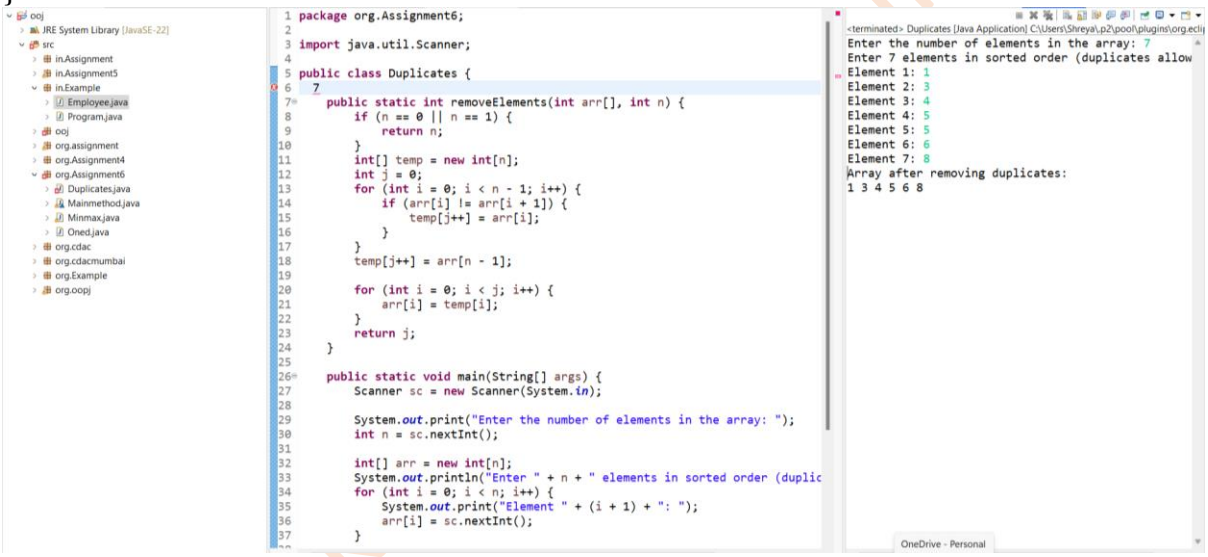
```
package org.Assignment6;
import java.util.Scanner;
public class Minmax {
		public static int findMax(int[] arr) {
			int max = arr[0];
			for(int i = 1; i < arr.length; i++) {
				if (arr[i] > max) {
					max = arr[i];
				}
			}
			return max;
		}
		public static int findMin(int[] arr) {
			int 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);

		 int[] arr = new int[5];
		 System.out.println("Enter 5 numbers:    ");
		 for (int i = 0; i < arr.length; i++) {
			 System.out.println("Enter Number " + (i + 1) + " : ");
			 arr[i] = sc.nextInt();

			 int max = findMax(arr);
			 int min = findMin(arr);
			 System.out.println("Maximum Value in the array: " +max);
			 System.out.println("Minimum Value in the array: " +min);

		 }
	}
}
```



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

```
package org.Assignment6;

import java.util.Scanner;

public class Duplicates {

    public static int removeElements(int arr[], int n) {
        if (n == 0 || n == 1) {
            return n;
        }
        int[] temp = new int[n];
        int j = 0;
        for (int i = 0; i < n - 1; i++) {
            if (arr[i] != arr[i + 1]) {
                temp[j++] = arr[i];
            }
        }
        temp[j++] = arr[n - 1];

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

    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();
```

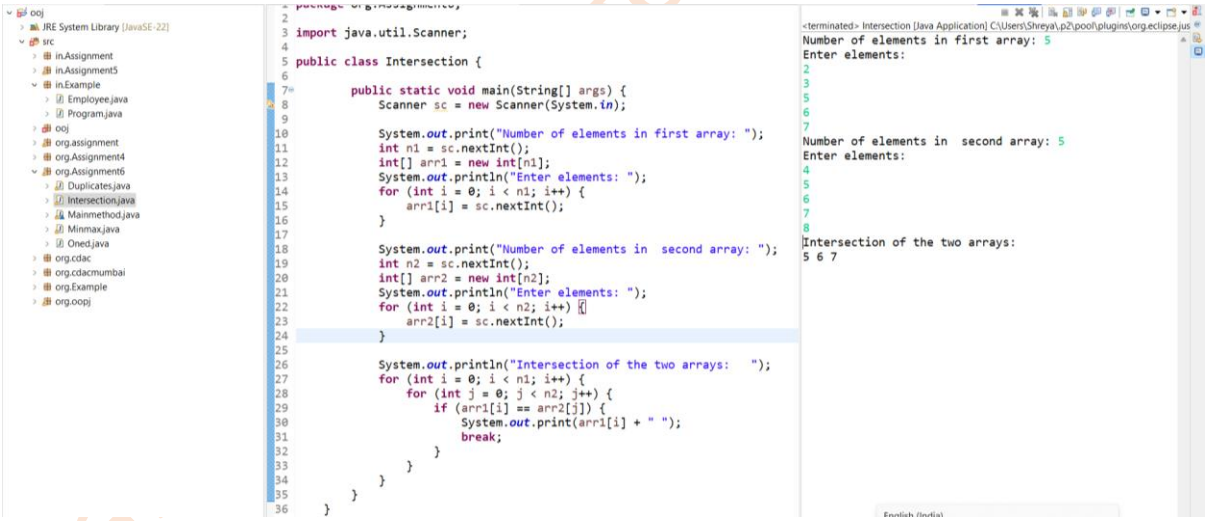
```
        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements in sorted order (duplicates allowed):");
        for (int i = 0; i < n; i++) {
            System.out.print("Element " + (i + 1) + ": ");
            arr[i] = sc.nextInt();
        }

        int length = removeElements(arr, n);

        System.out.println("Array after removing duplicates:");
        for (int i = 0; i < length; i++) {
          System.out.print(arr[i] + " ");
        }
    }
}
```



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

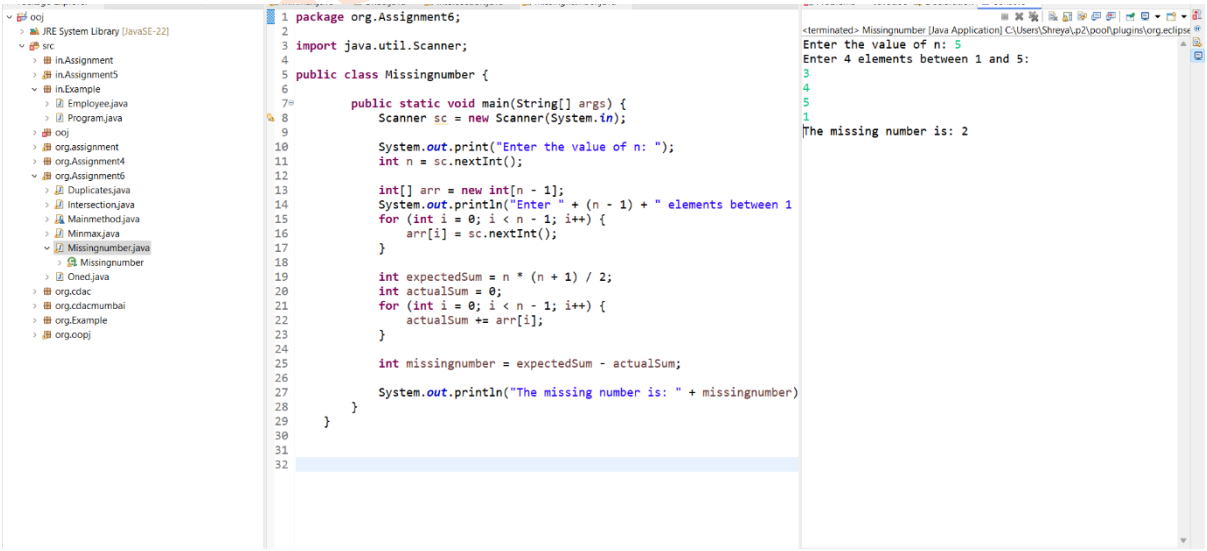
```
package org.Assignment6;
import java.util.Scanner;
public class Intersection {

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

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

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

        System.out.println("Intersection of the two arrays:   ");
        for (int i = 0; i < n1; i++) {
           for (int j = 0; j < n2; j++) {
              if (arr1[i] == arr2[j]) {
                 System.out.print(arr1[i] + " ");
                 break;
              }
           }
        }
     }
  }
```



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

```
package org.Assignment6;
import java.util.Scanner;
public class Missingnumber {

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

           System.out.print("Enter the value of n: ");
           int n = sc.nextInt();

           int[] arr = new int[n - 1];
           System.out.println("Enter " + (n - 1) + " elements between 1 and " + n + ":");
           for (int i = 0; i < n - 1; i++) {
              arr[i] = sc.nextInt();
           }

           int expectedSum = n * (n + 1) / 2;
           int actualSum = 0;
           for (int i = 0; i < n - 1; i++) {
              actualSum += arr[i];
           }

           int missingnumber = expectedSum - actualSum;

           System.out.println("The missing number is: " + missingnumber);
        }
     }
```



7. 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.

```
package org.Assignment6;
import java.util.Scanner;

class Assignment {
    private int[] arr;

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

    public void acceptRecord() {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter " + arr.length + "numbers: ");
      for (int i = 0; i < arr.length; i++) {
                        System.out.println("Number" + (i + 1) + " : ");
                        arr[i] = sc.nextInt();
                }
    }
      public void printRecord() {
                System.out.println("The no. elements in the array are:");
                for (int i = 0; i < arr.length; i++) {
                        System.out.println("Number" + (i + 1) + " : " + arr[i]);

                }
    }
}
 class Program1 {
      public static void main(String[] args) {
                Assignment ass = new Assignment(5);
                ass.acceptRecord();
                ass.printRecord();

                }
 }
```

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

9. **You need to implement a system to manage airplane seat assignments. The airplane has seats arranged in rows and columns. Implement functionalities to:**

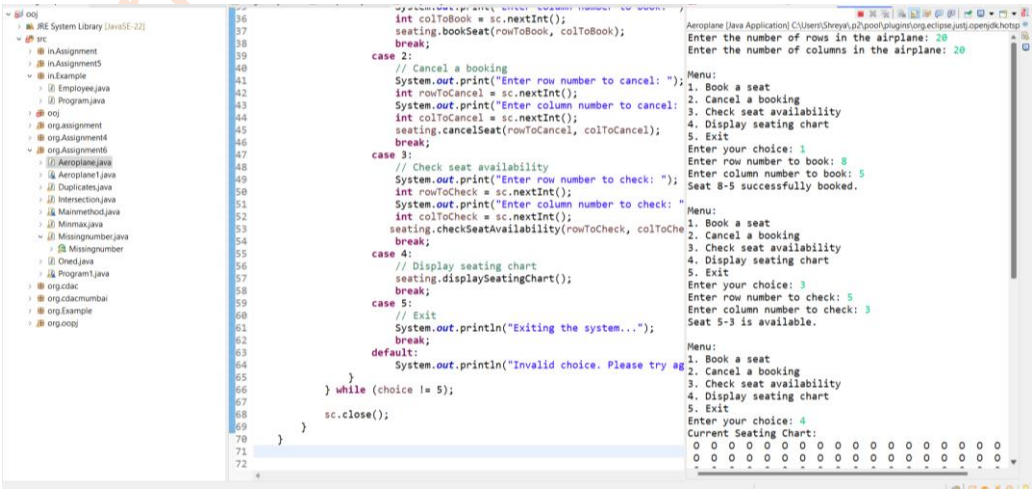
- **Initialize the seating arrangement with a given number of rows and columns.**
- **Book a seat to mark it as occupied.**
- **Cancel a booking to mark a seat as available.**
- **Check seat availability to determine if a specific seat is available.**
- **Display the current seating chart.**

```
package org.Assignment6;
class Aeroplane1 {
        private boolean[][] seats;
        private int rows;
        private int cols;

        public Aeroplane1(int rows, int cols) {
          this.rows = rows;
          this.cols = cols;
          seats = new boolean[rows][cols];
          }
          public boolean bookSeat(int row, int col) {
            if (isValidSeat(row, col)) {
              if (!seats[row - 1][col - 1]) { // if the seat is available
                seats[row - 1][col - 1] = true; // book the seat
                System.out.println("Seat " + row + "-" + col + " successfully booked.");
                return true;
              } else {
                System.out.println("Seat " + row + "-" + col + " is already occupied.");
                return false;
```

```
        }
    }
    return false;
}

// Method to cancel a seat booking
public boolean cancelSeat(int row, int col) {
    if (isValidSeat(row, col)) {
        if (seats[row - 1][col - 1]) { // if the seat is booked
            seats[row - 1][col - 1] = false; // cancel the booking
            System.out.println("Seat " + row + "-" + col + " booking canceled.");
            return true;
        } else {
            System.out.println("Seat " + row + "-" + col + " is already available.");
            return false;
        }
    }
    return false;
}

// Method to check if a seat is available
public boolean checkSeatAvailability(int row, int col) {
    if (isValidSeat(row, col)) {
        if (!seats[row - 1][col - 1]) {
            System.out.println("Seat " + row + "-" + col + " is available.");
            return true;
        } else {
            System.out.println("Seat " + row + "-" + col + " is occupied.");
            return false;
        }
    }
    return false;
}

// Method to display the seating chart
public void displaySeatingChart() {
    System.out.println("Current Seating Chart:");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (seats[i][j]) {
                System.out.print(" X "); // X indicates occupied seat
            } else {
                System.out.print(" O "); // O indicates available seat
            }
        }
        System.out.println();
    }
```

```
        }

        // Helper method to validate if a seat exists
        private boolean isValidSeat(int row, int col) {
            if (row < 1 || row > rows || col < 1 || col > cols) {
                System.out.println("Invalid seat number. Please try again.");
                return false;
            }
            return true;
        }
    }
package org.Assignment6;
import java.util.Scanner;
public class Aeroplane {

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

            System.out.print("Enter the number of rows in the airplane: ");
            int rows = sc.nextInt();

            System.out.print("Enter the number of columns in the airplane: ");
            int cols = sc.nextInt();

            // Initialize airplane seating
            Aeroplane1 seating = new Aeroplane1(rows, cols);

            int choice;
            do {
                System.out.println("\nMenu:");
                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: ");
                choice = sc.nextInt();

                switch (choice) {
                    case 1:
                        // Book a seat
                        System.out.print("Enter row number to book: ");
                        int rowToBook = sc.nextInt();
                        System.out.print("Enter column number to book: ");
                        int colToBook = sc.nextInt();
                        seating.bookSeat(rowToBook, colToBook);
                        break;
```

```
                case 2:
                    // Cancel a booking
                    System.out.print("Enter row number to cancel: ");
                    int rowToCancel = sc.nextInt();
                    System.out.print("Enter column number to cancel: ");
                    int colToCancel = sc.nextInt();
                    seating.cancelSeat(rowToCancel, colToCancel);
                    break;
                case 3:
                    // Check seat availability
                    System.out.print("Enter row number to check: ");
                    int rowToCheck = sc.nextInt();
                    System.out.print("Enter column number to check: ");
                    int colToCheck = sc.nextInt();
                   seating.checkSeatAvailability(rowToCheck, colToCheck);
                    break;
                case 4:
                    // Display seating chart
                    seating.displaySeatingChart();
                    break;
                case 5:
                    // Exit
                    System.out.println("Exiting the system...");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 5);

        sc.close();
    }
}
```



ASSIGNMENT NO.6