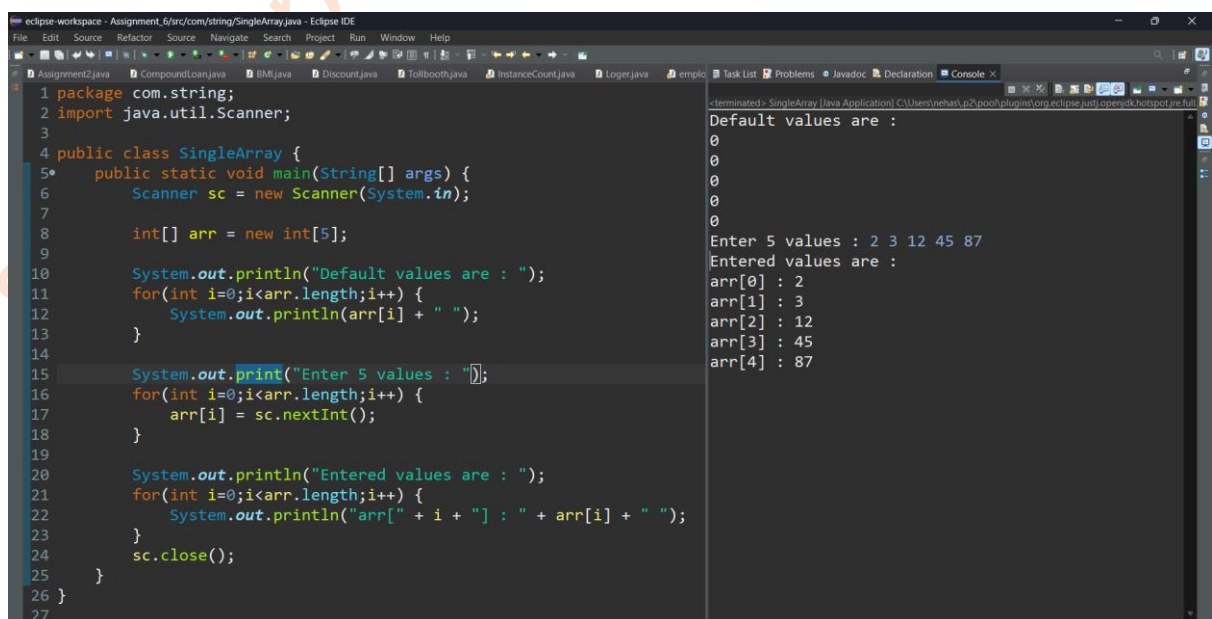


ASSIGNMENT NO.7

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 com.string;  
import java.util.Scanner;
```

```
public class SingleArray {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        int[] arr = new int[5];  
  
        System.out.println("Default values are : ");  
        for(int i=0;i<arr.length;i++) {  
            System.out.println(arr[i] + " ");  
        }  
  
        System.out.print("Enter 5 values : ");  
        for(int i=0;i<arr.length;i++) {  
            arr[i] = sc.nextInt();  
        }  
  
        System.out.println("Entered values are : ");  
        for(int i=0;i<arr.length;i++) {  
            System.out.println("arr[" + i + "] : " + arr[i] + " ");  
        }  
        sc.close();  
    }  
}
```



The screenshot shows the Eclipse IDE with a Java project named 'Assignment_6/src/com/string/SingleArray.java'. The code is as follows:

```
1 package com.string;  
2 import java.util.Scanner;  
3  
4 public class SingleArray {  
5     public static void main(String[] args) {  
6         Scanner sc = new Scanner(System.in);  
7  
8         int[] arr = new int[5];  
9  
10        System.out.println("Default values are : ");  
11        for(int i=0;i<arr.length;i++) {  
12            System.out.println(arr[i] + " ");  
13        }  
14  
15        System.out.print("Enter 5 values : ");  
16        for(int i=0;i<arr.length;i++) {  
17            arr[i] = sc.nextInt();  
18        }  
19  
20        System.out.println("Entered values are : ");  
21        for(int i=0;i<arr.length;i++) {  
22            System.out.println("arr[" + i + "] : " + arr[i] + " ");  
23        }  
24        sc.close();  
25    }  
26 }  
27
```

The console output shows the execution results:

```
<terminated> SingleArray [Java Application] C:\Users\nehas\p2\pool\plugins\org.eclipse.jdt.ui\org.eclipse.jdt.ui.hotspot.jre.full  
Default values are :  
0  
0  
0  
0  
0  
Enter 5 values : 2 3 12 45 87  
Entered values are :  
arr[0] : 2  
arr[1] : 3  
arr[2] : 12  
arr[3] : 45  
arr[4] : 87
```

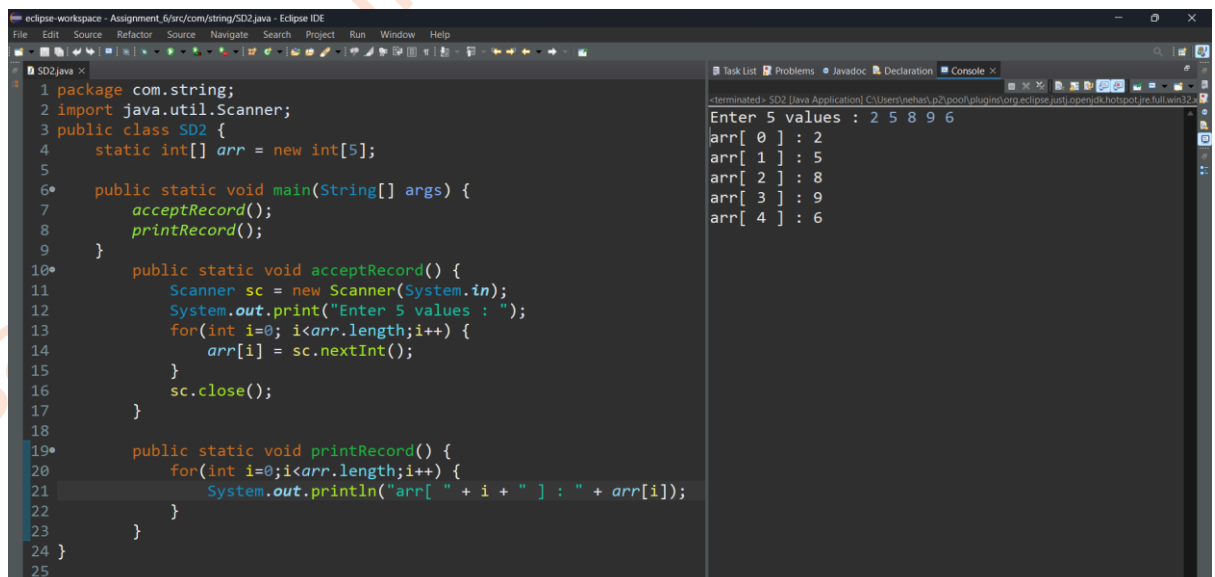
2. Declare a single-dimensional array of 5 integers inside the `main` method. Define a method named `acceptRecord` to get input from the terminal into the array and another method named `printRecord` to print the state of the array to the terminal.

```
package com.string;
import java.util.Scanner;
public class SD2 {
    static int[] arr = new int[5];

    public static void main(String[] args) {
        acceptRecord();
        printRecord();
    }

    public static void acceptRecord() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 5 values : ");
        for(int i=0; i<arr.length;i++) {
            arr[i] = sc.nextInt();
        }
        sc.close();
    }

    public static void printRecord() {
        for(int i=0;i<arr.length;i++) {
            System.out.println("arr[ " + i + " ] : " + arr[i]);
        }
    }
}
```



The screenshot shows the Eclipse IDE with the file `SD2.java` open. The code is as follows:

```
1 package com.string;
2 import java.util.Scanner;
3 public class SD2 {
4     static int[] arr = new int[5];
5
6     public static void main(String[] args) {
7         acceptRecord();
8         printRecord();
9     }
10    public static void acceptRecord() {
11        Scanner sc = new Scanner(System.in);
12        System.out.print("Enter 5 values : ");
13        for(int i=0; i<arr.length;i++) {
14            arr[i] = sc.nextInt();
15        }
16        sc.close();
17    }
18
19    public static void printRecord() {
20        for(int i=0;i<arr.length;i++) {
21            System.out.println("arr[ " + i + " ] : " + arr[i]);
22        }
23    }
24 }
25
```

The console output shows the program execution:

```
Enter 5 values : 2 5 8 9 6
arr[ 0 ] : 2
arr[ 1 ] : 5
arr[ 2 ] : 8
arr[ 3 ] : 9
arr[ 4 ] : 6
```

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

```
package com.string;
import java.util.Scanner;
```

```

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

        int[] a = new int[3];

        System.out.print("Enter elements : ");
        for(int i=0;i<a.length;i++) {
            a[i] = sc.nextInt();
        }

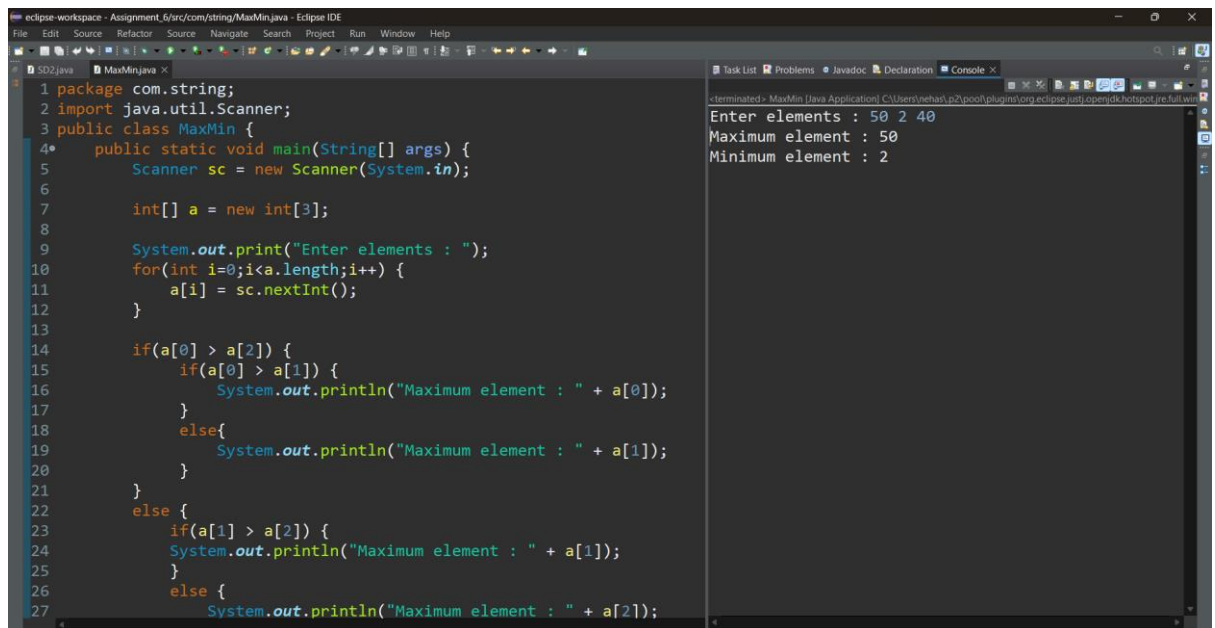
        if(a[0] > a[2]) {
            if(a[0] > a[1]) {
                System.out.println("Maximum element : " + a[0]);
            }
            else{
                System.out.println("Maximum element : " + a[1]);
            }
        }
        else {
            if(a[1] > a[2]) {
                System.out.println("Maximum element : " + a[1]);
            }
            else {
                System.out.println("Maximum element : " + a[2]);
            }
        }

        if(a[0] < a[2]) {
            if(a[0] < a[1]) {
                System.out.println("Minimum element : " + a[0]);
            }
            else{
                System.out.println("Minimum element : " + a[1]);
            }
        }
        else {
            if(a[1] < a[2]) {
                System.out.println("Minimum element : " + a[1]);
            }
            else {
                System.out.println("Minimum element : " + a[2]);
            }
        }

        sc.close();
    }
}

```

ASSIGNMENT NO.7



```
1 package com.string;
2 import java.util.Scanner;
3 public class MaxMin {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         int[] a = new int[3];
8
9         System.out.print("Enter elements : ");
10        for(int i=0;i<a.length;i++) {
11            a[i] = sc.nextInt();
12        }
13
14        if(a[0] > a[2]) {
15            if(a[0] > a[1]) {
16                System.out.println("Maximum element : " + a[0]);
17            }
18            else{
19                System.out.println("Maximum element : " + a[1]);
20            }
21        }
22        else {
23            if(a[1] > a[2]) {
24                System.out.println("Maximum element : " + a[1]);
25            }
26            else {
27                System.out.println("Maximum element : " + a[2]);
28            }
29        }
30    }
31}
```

Console Output:

```
Enter elements : 50 2 40
Maximum element : 50
Minimum element : 2
```

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

```
package com.string;
```

```
public class Duplicate {
```

```
    public static void main(String[] args) {
        int[] a = { 4, 3, 4, 9, 3, 5, 6, 1, 8 };
        int n = a.length;
        int m = 0;
```

```
        for (int i = 0; i < n; i++) {
            m = Math.max(m, a[i]);
        }
```

```
        int[] f = new int[m + 1];
```

```
        for (int i = 0; i < n; i++) {
            f[a[i]]++;
        }
```

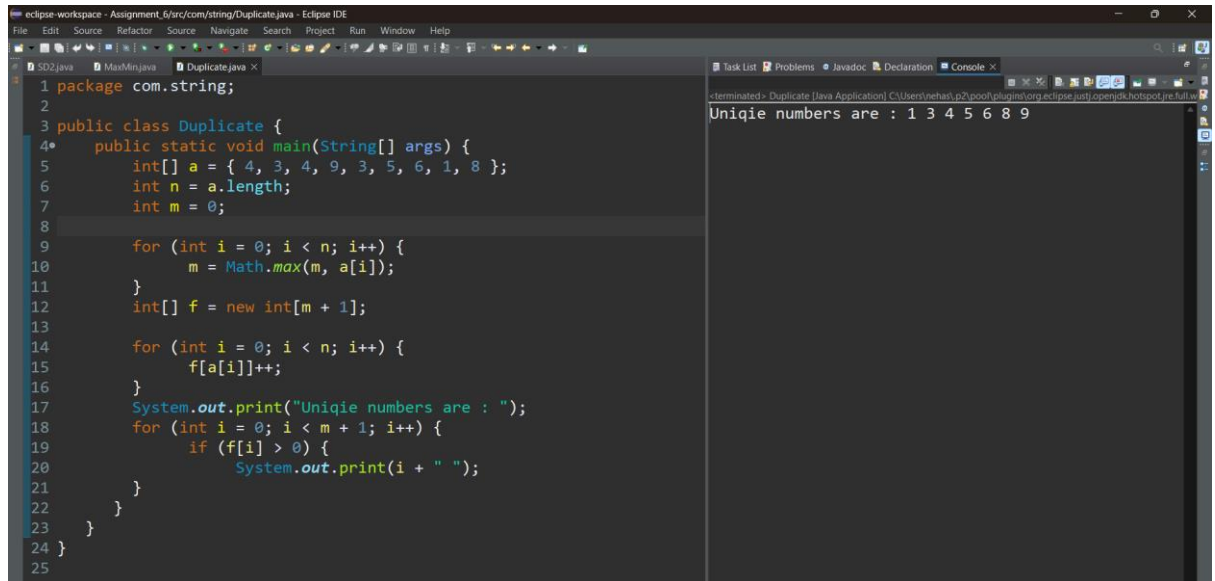
```
        System.out.print("Unique numbers are : ");
```

```
        for (int i = 0; i < m + 1; i++) {
```

```
            if (f[i] > 0) {
                System.out.print(i + " ");
            }
        }
```

```
    }
}
```

ASSIGNMENT NO.7



```
1 package com.string;
2
3 public class Duplicate {
4     public static void main(String[] args) {
5         int[] a = { 4, 3, 4, 9, 3, 5, 6, 1, 8 };
6         int n = a.length;
7         int m = 0;
8
9         for (int i = 0; i < n; i++) {
10             m = Math.max(m, a[i]);
11         }
12         int[] f = new int[m + 1];
13
14         for (int i = 0; i < n; i++) {
15             f[a[i]]++;
16         }
17         System.out.print("Unique numbers are : ");
18         for (int i = 0; i < m + 1; i++) {
19             if (f[i] > 0) {
20                 System.out.print(i + " ");
21             }
22         }
23     }
24 }
25
```

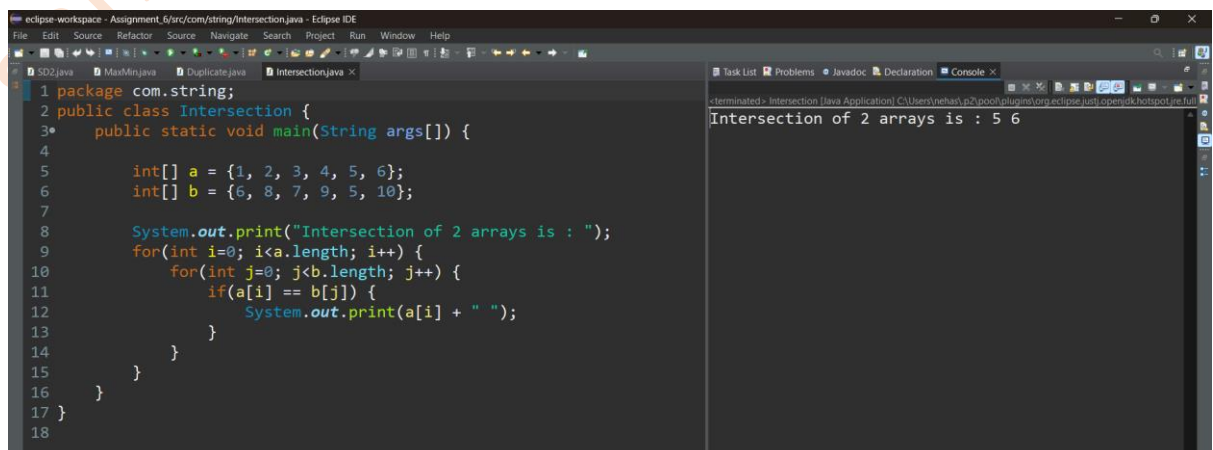
Unique numbers are : 1 3 4 5 6 8 9

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

```
package com.string;
public class Intersection {
    public static void main(String args[]) {

        int[] a = { 1, 2, 3, 4, 5, 6 };
        int[] b = { 6, 8, 7, 9, 5, 10 };

        System.out.print("Intersection of 2 arrays is : ");
        for(int i=0; i<a.length; i++) {
            for(int j=0; j<b.length; j++) {
                if(a[i] == b[j]) {
                    System.out.print(a[i] + " ");
                }
            }
        }
    }
}
```



```
1 package com.string;
2 public class Intersection {
3     public static void main(String args[]) {
4
5         int[] a = { 1, 2, 3, 4, 5, 6 };
6         int[] b = { 6, 8, 7, 9, 5, 10 };
7
8         System.out.print("Intersection of 2 arrays is : ");
9         for(int i=0; i<a.length; i++) {
10             for(int j=0; j<b.length; j++) {
11                 if(a[i] == b[j]) {
12                     System.out.print(a[i] + " ");
13                 }
14             }
15         }
16     }
17 }
18
```

Intersection of 2 arrays is : 5 6

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

```
package com.string;
```

```
public class MissingNum {
```

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

```
        int[] x = {1, 2, 3, 5};
```

```
        int n = 5;
```

```
        int sum = 0;
```

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

```
            sum += x[i];
```

```
        }
```

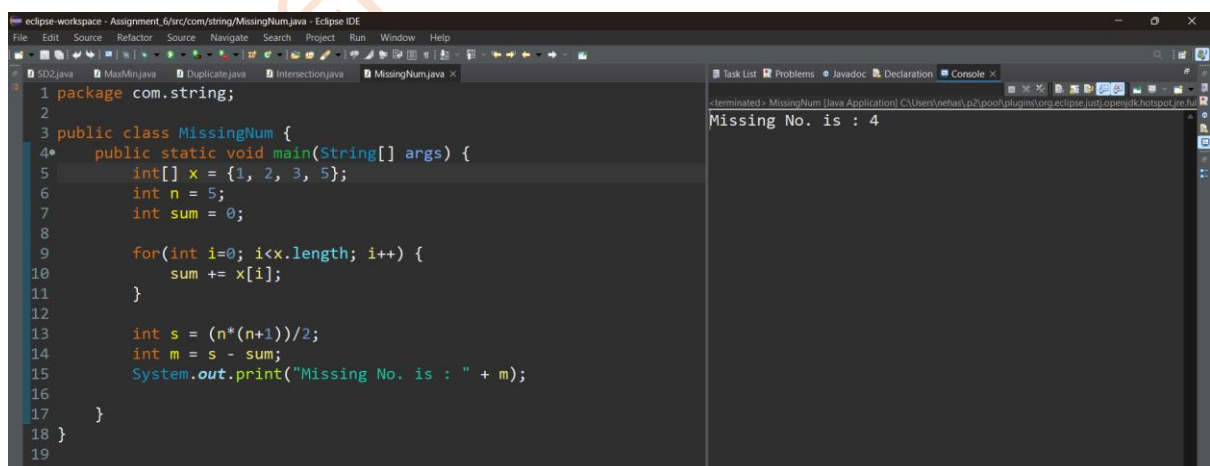
```
        int s = (n*(n+1))/2;
```

```
        int m = s - sum;
```

```
        System.out.print("Missing No. is : " + m);
```

```
    }
```

```
}
```



```
1 package com.string;
2
3 public class MissingNum {
4     public static void main(String[] args) {
5         int[] x = {1, 2, 3, 5};
6         int n = 5;
7         int sum = 0;
8
9         for(int i=0; i<x.length; i++) {
10             sum += x[i];
11         }
12
13         int s = (n*(n+1))/2;
14         int m = s - sum;
15         System.out.print("Missing No. is : " + m);
16     }
17 }
18
19
```

Task List | Problems | Javadoc | Declaration | Console x

<terminated> MissingNum [Java Application] C:\Users\neha\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full

Missing No. is : 4

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 com.string;

public class Q7 {

    private int[] a;

    Q7(int s){
        a = new int[s];
    }

    public void acceptRecord(int i, int v) {
        if(i>=0 && i<a.length) {
            a[i] = v;
        }
        else {
            System.out.println("Invalid!!!!");
        }
    }

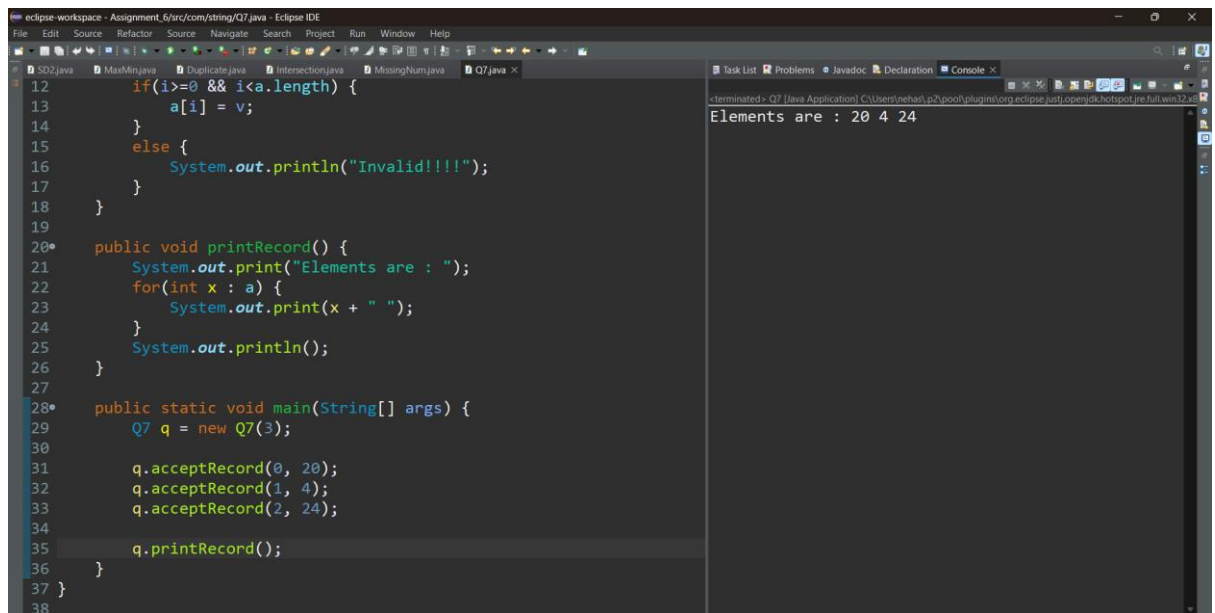
    public void printRecord() {
        System.out.print("Elements are : ");
        for(int x : a) {
            System.out.print(x + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Q7 q = new Q7(3);

        q.acceptRecord(0, 20);
        q.acceptRecord(1, 4);
        q.acceptRecord(2, 24);

        q.printRecord();
    }
}
```

ASSIGNMENT NO.7



```
12      if(i>=0 && i<a.length) {
13          a[i] = v;
14      }
15      else {
16          System.out.println("Invalid!!!!");
17      }
18  }
19
20  public void printRecord() {
21      System.out.print("Elements are : ");
22      for(int x : a) {
23          System.out.print(x + " ");
24      }
25      System.out.println();
26  }
27
28  public static void main(String[] args) {
29      Q7 q = new Q7(3);
30
31      q.acceptRecord(0, 20);
32      q.acceptRecord(1, 4);
33      q.acceptRecord(2, 24);
34
35      q.printRecord();
36  }
37 }
38
```

Console Output:
Elements are : 20 4 24

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

```
package com.string;
```

```
public class GetSet {
```

```
    private int[] a;
```

```
    public GetSet(int s) {
        a = new int[s];
    }
```

```
    public void setA(int i, int v) {
        if(i>=0 && i<a.length) {
            a[i] = v;
        }
        else {
            System.out.println("Index out of bound.....");
        }
    }
```

```
    public int getA(int i) {
        if(i>=0 && i<a.length) {
            return a[i];
        }
        else {
            System.out.println("Index out of bound.....");
            return 0;
        }
    }
}
```



```

public void printRecord() {
    System.out.print("Elements are : ");
    for(int x : a) {
        System.out.print(x + " ");
    }
    System.out.println();
}

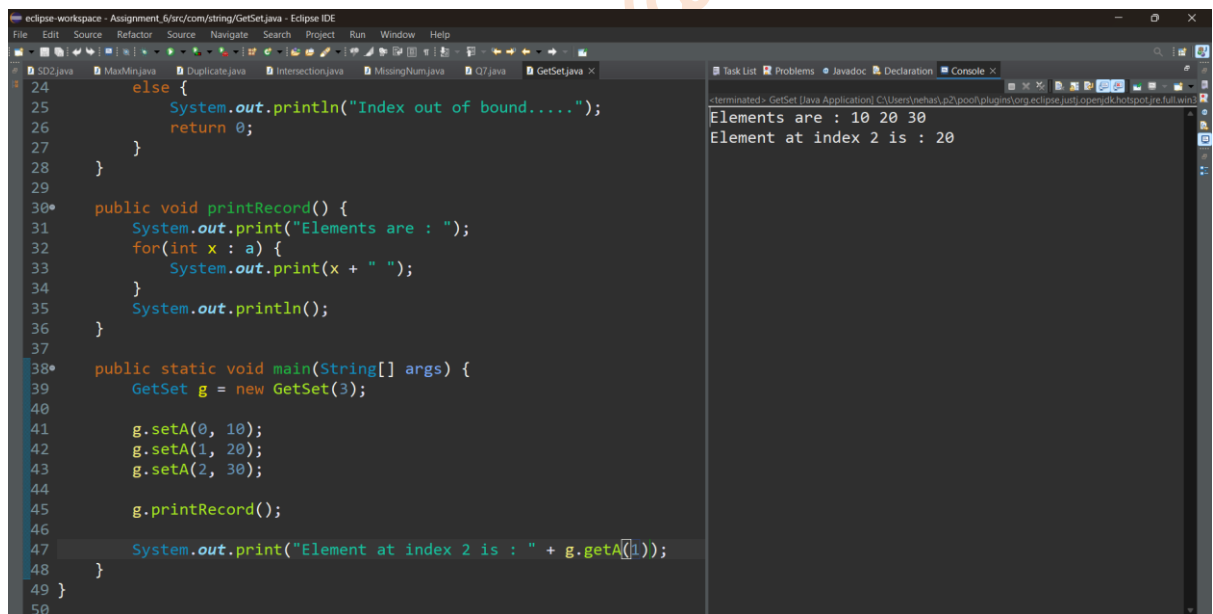
public static void main(String[] args) {
    GetSet g = new GetSet(3);

    g.setA(0, 10);
    g.setA(1, 20);
    g.setA(2, 30);

    g.printRecord();

    System.out.print("Element at index 2 is : " + g.getA(1));
}
}

```



The screenshot shows the Eclipse IDE with a Java file named 'GetSet.java'. The code is as follows:

```

24 else {
25     System.out.println("Index out of bound.....");
26     return 0;
27 }
28 }
29
30 public void printRecord() {
31     System.out.print("Elements are : ");
32     for(int x : a) {
33         System.out.print(x + " ");
34     }
35     System.out.println();
36 }
37
38 public static void main(String[] args) {
39     GetSet g = new GetSet(3);
40
41     g.setA(0, 10);
42     g.setA(1, 20);
43     g.setA(2, 30);
44
45     g.printRecord();
46
47     System.out.print("Element at index 2 is : " + g.getA(1));
48 }
49 }
50

```

The console output on the right shows:

```

Elements are : 10 20 30
Element at index 2 is : 20

```

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 com.string;
```

```
public class PlaneSeating {  
  
    private char[][] seat;  
  
    private int row;  
  
    private int col;  
  
    public PlaneSeating(int row, int col) {  
  
        this.row = row;  
  
        this.col = col;  
  
        seat = new char[row][col];  
  
        for(int i=0; i<row; i++) {  
            for(int j=0; j<col; j++) {  
                seat[i][j] = 'A';  
            }  
        }  
    }  
  
    public void bookSeat(int row, int col) {  
  
        if(seat[row][col] == 'A') {  
            seat[row][col] = 'x';  
  
            System.out.println("Seat book sucessfully.");  
  
        }else {  
  
            System.out.println("Seat not available.");  
  
        }  
    }  
}
```

```
public void cancelSeat(int row, int col) {  
    if(seat[row][col] == 'x') {  
        seat[row][col] = 'A';  
        System.out.println("Seat cancelled sucessfully.");  
    }else {  
        System.out.println("Seat is available.");  
    }  
}
```

```
public void displayRecord() {  
    System.out.println("Seats available : ");  
    for(int i=0; i<row; i++) {  
        for(int j=0; j<col; j++) {  
            System.out.print(seat[i][j]);  
        }  
        System.out.println();  
    }  
}
```

```
public static void main(String[] args) {  
    PlaneSeating p = new PlaneSeating(4, 4);  
    p.bookSeat(2,2);  
    p.displayRecord();  
}
```

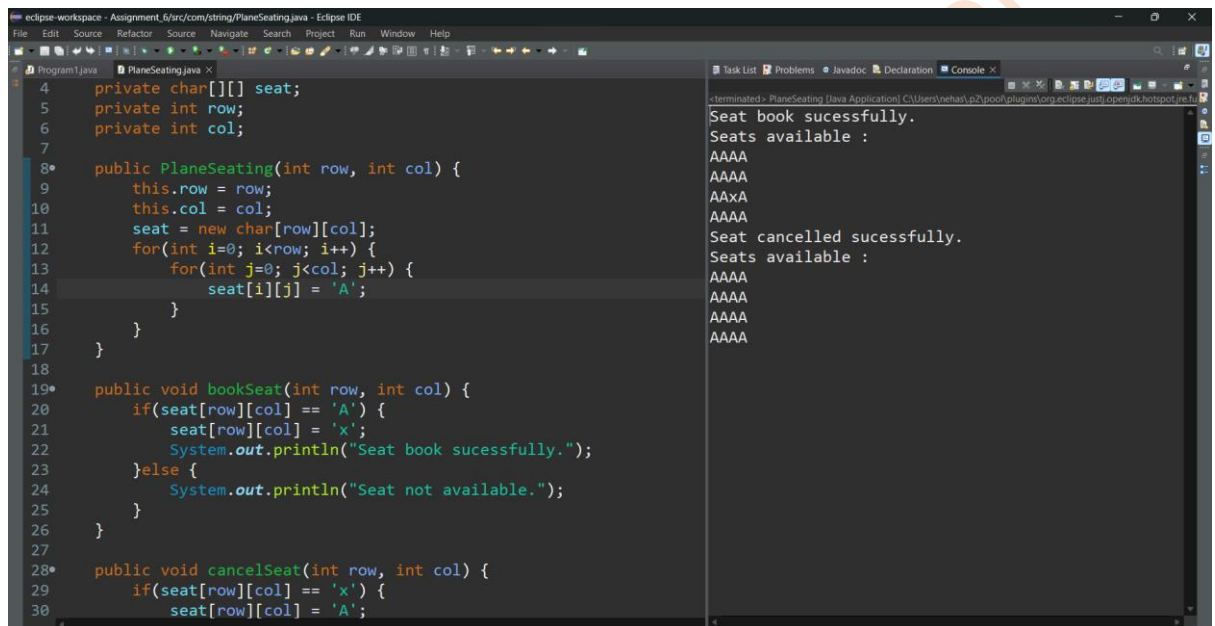
ASSIGNMENT NO.7

```
        p.cancelSeat(2,2);

        p.displayRecord();

    }

}
```



The screenshot shows the Eclipse IDE with a Java project named 'Assignment_5'. The main editor displays the 'PlaneSeating.java' file. The code defines a class with private attributes 'seat' (a 2D char array), 'row', and 'col'. It includes three methods: 'PlaneSeating' (constructor), 'bookSeat', and 'cancelSeat'. The 'bookSeat' method checks if a seat is available (marked 'A') and marks it as booked (marked 'X') if successful. The 'cancelSeat' method checks if a seat is booked (marked 'X') and marks it as available (marked 'A') if successful. The console on the right shows the output of the program, including the initial state of the seat matrix, the successful booking of seat (2,2), and the resulting seat matrix after the booking.

```
4 private char[][] seat;
5 private int row;
6 private int col;
7
8 public PlaneSeating(int row, int col) {
9     this.row = row;
10    this.col = col;
11    seat = new char[row][col];
12    for(int i=0; i<row; i++) {
13        for(int j=0; j<col; j++) {
14            seat[i][j] = 'A';
15        }
16    }
17 }
18
19 public void bookSeat(int row, int col) {
20     if(seat[row][col] == 'A') {
21         seat[row][col] = 'X';
22         System.out.println("Seat book sucessfully.");
23     }else {
24         System.out.println("Seat not available.");
25     }
26 }
27
28 public void cancelSeat(int row, int col) {
29     if(seat[row][col] == 'X') {
30         seat[row][col] = 'A';
31     }
32 }
```

Task List Problems Javadoc Declaration Console

```
<terminated> PlaneSeating [Java Application] C:\Users\nehar\p2\workspace\Assignment_5\PlaneSeating.java
Seat book sucessfully.
Seats available :
AAAA
AAAA
AAXA
AAAA
Seat cancelled sucessfully.
Seats available :
AAAA
AAAA
AAAA
AAAA
```