

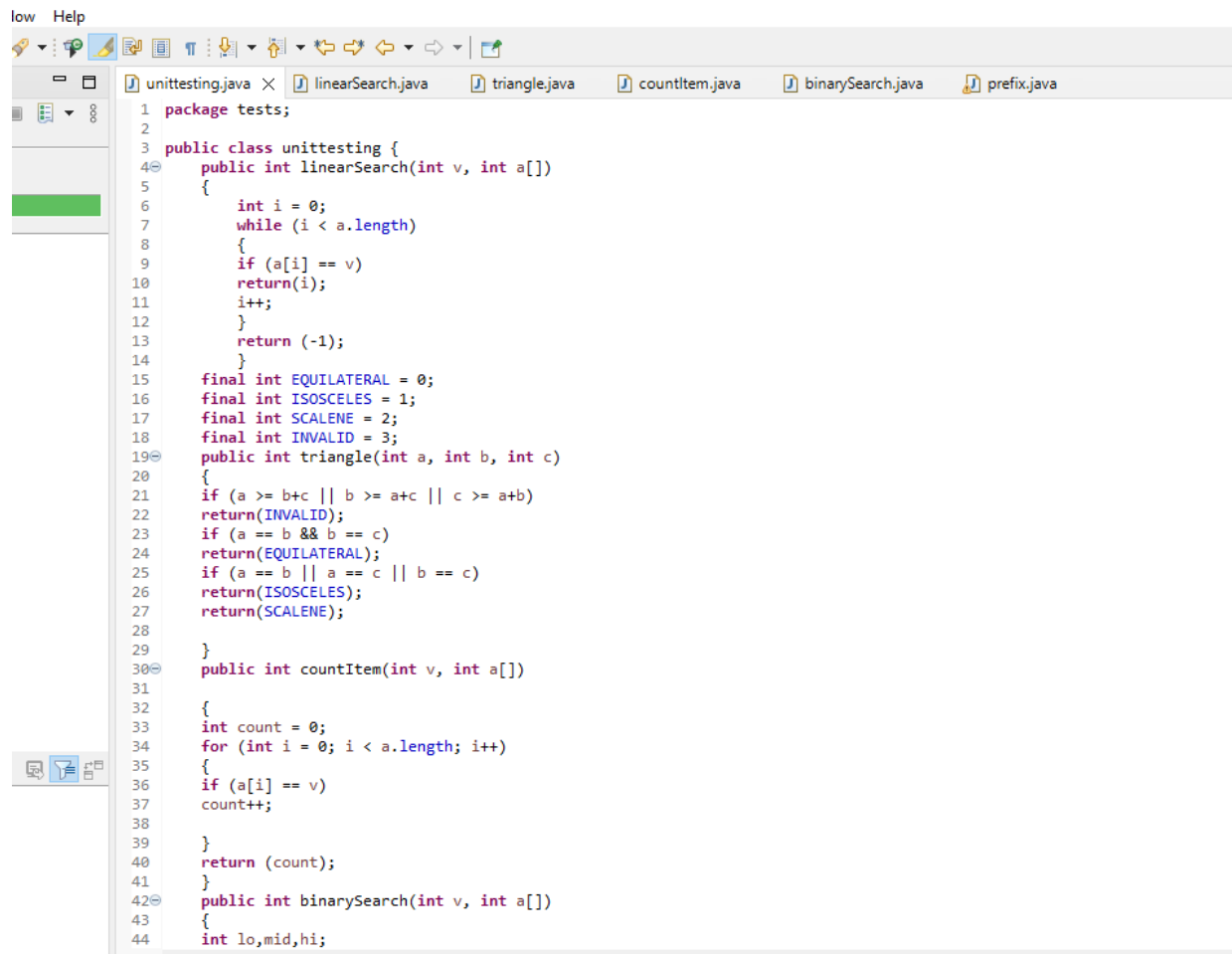
IT314 Software Engineering

Lab 7

Name: Noopur Chaudhary

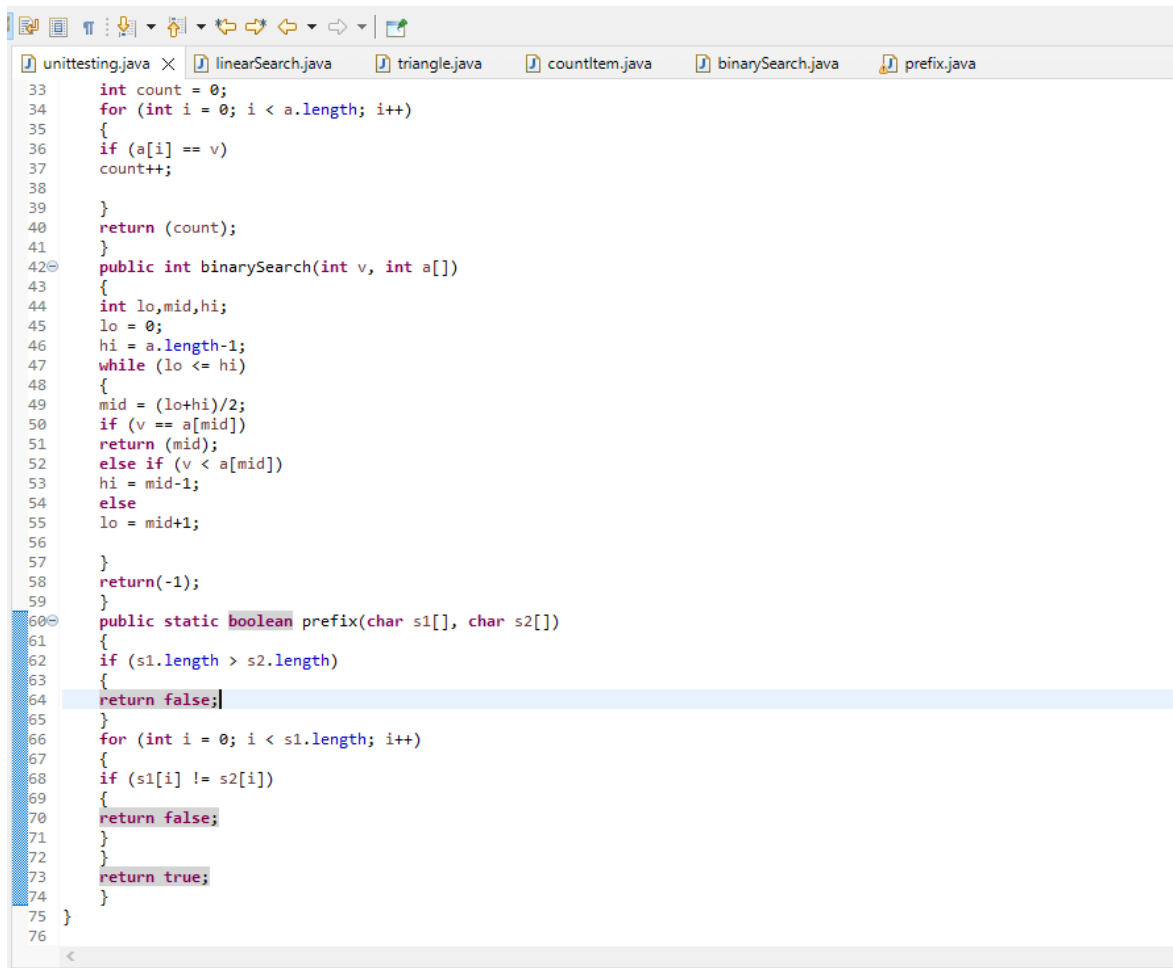
ID: 202001091

Code:



The screenshot shows an IDE window with a menu bar (low, Help) and a toolbar. The tab bar at the top lists several files: unittesting.java, linearSearch.java, triangle.java, countItem.java, binarySearch.java, and prefix.java. The main editor area displays the code for unittesting.java, which is a Java class for testing various algorithms. The code includes package declarations, class definitions, and several methods: linearSearch, triangle, countItem, and binarySearch. The linearSearch method uses a while loop to find the index of a value in an array. The triangle method uses conditional logic to determine the type of triangle based on side lengths. The countItem method uses a for loop to count the occurrences of a value in an array. The binarySearch method is partially visible at the bottom.

```
1 package tests;
2
3 public class unittesting {
4     public int linearSearch(int v, int a[])
5     {
6         int i = 0;
7         while (i < a.length)
8         {
9             if (a[i] == v)
10                return(i);
11                i++;
12            }
13            return (-1);
14        }
15        final int EQUILATERAL = 0;
16        final int ISOSCELES = 1;
17        final int SCALENE = 2;
18        final int INVALID = 3;
19        public int triangle(int a, int b, int c)
20        {
21            if (a >= b+c || b >= a+c || c >= a+b)
22                return(INVALID);
23            if (a == b && b == c)
24                return(EQUILATERAL);
25            if (a == b || a == c || b == c)
26                return(ISOSCELES);
27            return(SCALENE);
28        }
29    }
30    public int countItem(int v, int a[])
31    {
32        int count = 0;
33        for (int i = 0; i < a.length; i++)
34        {
35            if (a[i] == v)
36                count++;
37        }
38        return (count);
39    }
40    public int binarySearch(int v, int a[])
41    {
42        int lo,mid,hi;
```



```
33     int count = 0;
34     for (int i = 0; i < a.length; i++)
35     {
36         if (a[i] == v)
37             count++;
38     }
39     return (count);
40 }
41 }
42 public int binarySearch(int v, int a[])
43 {
44     int lo,mid,hi;
45     lo = 0;
46     hi = a.length-1;
47     while (lo <= hi)
48     {
49         mid = (lo+hi)/2;
50         if (v == a[mid])
51             return (mid);
52         else if (v < a[mid])
53             hi = mid-1;
54         else
55             lo = mid+1;
56     }
57     return(-1);
58 }
59 }
60 public static boolean prefix(char s1[], char s2[])
61 {
62     if (s1.length > s2.length)
63     {
64         return false;
65     }
66     for (int i = 0; i < s1.length; i++)
67     {
68         if (s1[i] != s2[i])
69         {
70             return false;
71         }
72     }
73     return true;
74 }
75 }
76 }
```

Table for P1:

eclipse-workspace - JavaTesting/src/tests/linearSearch.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit ×

Finished after 0.012 seconds

Runs: 1/1 Errors: 0 Failures: 0

> tests.triangle [Runner: JUnit 4] (0.000 s)

```

1 package tests;
2
3 import static org.junit.Assert.*;
4
5
6 public class linearSearch {
7
8
9     @Test
10    public void test1() {
11        unittesting obj1= new unittesting();
12        int arry[]= {1,2,3,4,5};
13        int output_f= obj1.linearSearch(4, arry);
14        assertEquals(3,output_f);
15    }
16    public void test2() {
17        unittesting obj1= new unittesting();
18        int arry[]= {1,2,3,'a',5};
19        int output_f= obj1.linearSearch(4, arry);
20        assertEquals(-1,output_f);
21    }
22    public void test3() {
23        unittesting obj1= new unittesting();
24        int arry[]= {1,2,3,5};
25        int output_f= obj1.linearSearch(4, arry);
26        assertEquals(-1,output_f);
27    }
28    public void test4() {
29        unittesting obj1= new unittesting();
30        int arry[]= {'\0'};
31        int output_f= obj1.linearSearch(4, arry);
32        assertEquals(-1,output_f);
33    }
34 }
35

```

Values	Equivalent Output	Expected Output
4,{1,2,3,4}	3	3
4,{1,2,3,'a',5}	-1	-1
4, {1,2,3,5}	-1	-1
4, {'\0'}	-1	-1

Table for P2:

```

1 package tests;
2
3 import static org.junit.Assert.*;
4
5
6 public class triangle {
7
8
9     @Test
10    public void test1() {
11        unittesting obj1= new unittesting();
12        int output_f= obj1.triangle(4,4,4);
13        assertEquals(0,output_f);
14    }
15
16    public void test2() {
17        unittesting obj1= new unittesting();
18        int output_f= obj1.triangle(2,1,4);
19        assertEquals(3,output_f);
20    }
21
22    public void test3() {
23        unittesting obj1= new unittesting();
24        int output_f= obj1.triangle('a',4,4);
25        assertEquals(3,output_f);
26    }
27
28    public void test4() {
29        unittesting obj1= new unittesting();
30        int output_f= obj1.triangle(3,4,5);
31        assertEquals(2,output_f);
32    }
33
34    public void test5() {
35        unittesting obj1= new unittesting();
36        int output_f= obj1.triangle(3,4,4);
37        assertEquals(1,output_f);
38    }
39
40 }

```

Values	Equivalent Output	Expected Output
4, 4, 4	0	0
2, 1, 4	3	3
'a', 4, 4	3	3
3, 4, 5	2	2
3, 4, 4	1	1

Table for P3:

eclipse-workspace - JavaTesting/src/tests/countItem.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit ×

Finished after 0.022 seconds

Runs: 1/1 Errors: 0 Failures: 0

> tests.countItem [Runner: JUnit 4] (0.002 s)

```

1 package tests;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class countItem {
8
9     @Test
10    public void test1() {
11        unittesting obj1= new unittesting();
12        int arry[]= {1,2,3,4,5};
13        int output_f= obj1.countItem(4, arry);
14        assertEquals(1,output_f);
15    }
16
17    public void test2() {
18        unittesting obj1= new unittesting();
19        int arry[]= {1,2,3,'a',5};
20        int output_f= obj1.countItem(4, arry);
21        assertEquals(0,output_f);
22    }
23
24    public void test3() {
25        unittesting obj1= new unittesting();
26        int arry[]= {4,4,4,4,4};
27        int output_f= obj1.countItem(4, arry);
28        assertEquals(5,output_f);
29    }
30
31    public void test4() {
32        unittesting obj1= new unittesting();
33        int arry[]= {'\0'};
34        int output_f= obj1.countItem(4, arry);
35        assertEquals(0,output_f);
36    }
37 }

```

Values	Equivalent Output	Expected Output
4, {1, 2, 3, 4, 5}	1	1
4, {1, 2, 3, 'a', 5}	0	0
4, {4, 4, 4, 4, 4}	5	5
4, {'\0'}	0	0

Table for P4:

eclipse-workspace - JavaTesting/src/tests/binarySearch.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit ×

Finished after 0.018 seconds

Runs: 1/1 Errors: 0 Failures: 0

> tests.binarySearch [Runner: JUnit 4] (0.002 s)

```

1 package tests;
2
3 import static org.junit.Assert.*;
4
5
6 public class binarySearch {
7
8
9     @Test
10    public void test1() {
11        unittesting obj1= new unittesting();
12        int array[]= {1,2,3,4,5};
13        int output_f= obj1.linearSearch(4, array);
14        assertEquals(3,output_f);
15    }
16
17    public void test2() {
18        unittesting obj1= new unittesting();
19        int array[]= {1,2,3,8,5};
20        int output_f= obj1.linearSearch(-1, array);
21        assertEquals(3,output_f);
22    }
23
24    public void test3() {
25        unittesting obj1= new unittesting();
26        int array[]= {1,2,3,'a',5};
27        int output_f= obj1.linearSearch(-1, array);
28        assertEquals(3,output_f);
29    }
30
31    public void test4() {
32        unittesting obj1= new unittesting();
33        int array[]= {'\0'};
34        int output_f= obj1.linearSearch(-1, array);
35        assertEquals(3,output_f);
36    }
37 }

```

Values	Equivalent Output	Expected Output
4, {1, 2, 3, 4, 5}	3	1
-1, {1, 2, 3, 8, 5}	3	2
-1, {1, 2, 3, 'a', 5}	3	0
-1, {'\0'}	3	3

Table for P5:

eclipse-workspace - JavaTesting/src/tests/prefix.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit X

Finished after 0.021 seconds

Runs: 1/1 Errors: 0 Failures: 1

tests.prefix [Runner: JUnit 4] (0.005 s)

test1 (0.005 s)

Failure Trace

java.lang.AssertionError: expected:<1> but was:<true>

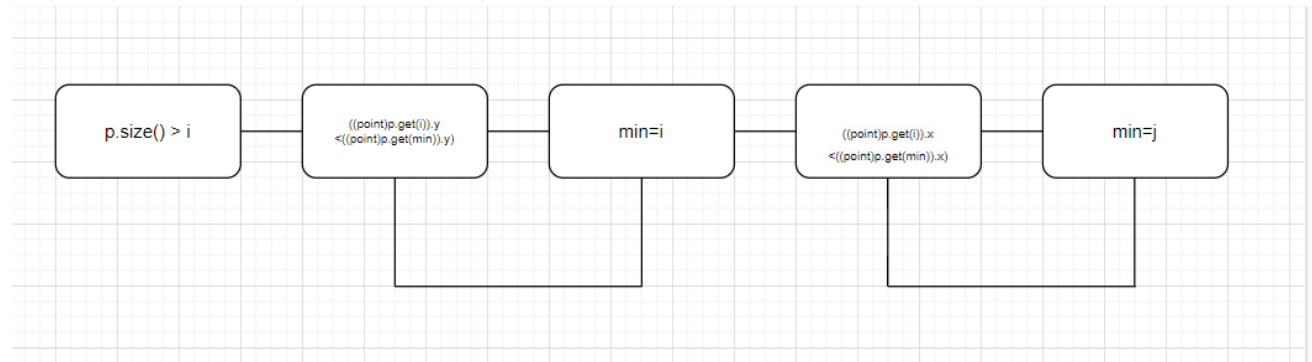
```

1 package tests;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class prefix {
8
9     @Test
10    public void test1() {
11        unittesting obj1= new unittesting();
12        char a[]= {'h','e'};
13        char b[]= {'h','e'};
14        boolean output_f= obj1.prefix(a, b);
15        assertEquals(1,output_f);
16    }
17
18    public void test2() {
19        unittesting obj1= new unittesting();
20        char a[]= {'e'};
21        char b[]= {'h','e'};
22        boolean output_f= obj1.prefix(a, b);
23        assertEquals(0,output_f);
24    }
25
26    public void test3() {
27        unittesting obj1= new unittesting();
28        char a[]= {'h','t'};
29        char b[]= {'h','e'};
30        boolean output_f= obj1.prefix(a, b);
31        assertEquals(0,output_f);
32    }
33
34    public void test4() {
35        unittesting obj1= new unittesting();
36        char a[]= {'h','e','d'};
37        char b[]= {'h','e',1};
38        boolean output_f= obj1.prefix(a, b);
39        assertEquals(0,output_f);
40    }
41
42 }

```

Values	Equivalent Output	Expected Output
{h, e}, {h, e}	1	1
{e}, {h, e}	0	0
{h, t}, {h, e}	0	0
{h, e, d}, {h, e, 1}	0	0

Section B



2. a. Statement Coverage test set:

Test Case 1: `p.size() > point` i.e. 2 is false

Test Case 2: `p.size() > 2` is true

b. Branch Coverage Test Set:

Test Case 1: `p.size() > point` i.e. 2 is false

Test Case 2: `p.size() > 2` is true and loop is executed

Test Case 3: `p.size() > 2` is true and loop is not executed

c. Basic Condition Coverage Test Set:

Test Case 1: `p.size() > point` i.e. 2 is false

Test Case 2: `p.size() > 2` is true and loop is executed

Test Case 3: `p.size() > 2` is true and loop is not executed

Test Case 4: `p.size() > 2` is true and loop is executed twice