

Name : Pansuriya Viraj Dhirajbhai

Student Id:202001196

LAB-8

Date:20/04/2023

Course:IT314

## Creating Boa Class:

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows the project structure with packages `lab8`, `lab8_202001196_viraj`, and `src`. The `src` package contains the `Boa.java` file.
- Editor:** Displays the `Boa.java` file with the following code:

```
1 package lab8_202001196_viraj;
2
3 public class Boa {
4     private String name;
5     private int length; // the length of the boa, in feet
6     private String favoriteFood;
7
8
9     public Boa (String name, int length, String favoriteFood){
10         this.name = name;
11         this.length = length;
12         this.favoriteFood = favoriteFood;
13     }
14
15     // returns true if this boa constructor is healthy
16     public boolean isHealthy(){
17         return this.favoriteFood.equals("granola bars");
18     }
19
20     // returns true if the length of this boa constructor is
21     // less than the given cage length
22     public boolean fitsInCage(int cageLength){
23         return this.length < cageLength;
24     }
25 }
26
```
- Outline:** Shows the class `Boa` with its attributes: `name: String`, `length: int`, `favoriteFood: String`, `isHealthy(): boolean`, and `fitsInCage(int): boolean`.
- Problems:** Empty table.
- Coverage:** Empty table.

## Creating Boa-test file

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows the project structure with packages `lab8`, `lab8_202001196_viraj`, and `src`. The `src` package contains the `BoaTest.java` file.
- Editor:** Displays the `BoaTest.java` file with the following code:

```
1 package lab8_202001196_viraj;
2
3 import static org.junit.Assert.*;
4 import org.junit.Before;
5 import org.junit.Test;
6
7 public class BoaTest {
8     private Boa jen, ken;
9
10
11     @Before
12     public void setUp() throws Exception {
13         jen = new Boa("Jennifer", 2, "grapes");
14         ken = new Boa("Kenneth", 3, "granola bars");
15     }
16
17     @Test
18     public void testIsHealthy() {
19         //
20     }
21
22     @Test
23     public void testFitsInCage() {
24         //
25     }
26 }
27
28
```
- Outline:** Shows the class `BoaTest` with its methods: `setUp(): void`, `testIsHealthy(): void`, and `testFitsInCage(): void`.
- Problems:** Empty table.
- Coverage:** Empty table.

Here , i have modify both methods,

```
1 package lab8_202001196_viraj;
2
3 import static org.junit.Assert.*;
4 import org.junit.Before;
5 import org.junit.Test;
6
7 public class BoaTest {
8     private Boa jen,ken;
9
10    @Before
11    public void setUp() throws Exception {
12        jen = new Boa("Jennifer", 2, "grapes");
13        ken = new Boa("Kenneth", 3, "granola bars");
14    }
15
16    @Test
17    public void testIsHealthy() {
18        assertEquals("Error in isHealthy()",false,jen.isHealthy());
19        assertEquals("Error in isHealthy()",true,jen.isHealthy());
20        assertEquals("Error in isHealthy()",false,ken.isHealthy());
21        assertEquals("Error in isHealthy()",true,ken.isHealthy());
22    }
23
24    @Test
25    public void testFitsInCage() {
26        assertEquals("Error in FitsInCage()",true,ken.fitsInCage(10));
27        assertEquals("Error in FitsInCage()",true,ken.fitsInCage(2));
28        assertEquals("Error in FitsInCage()",false,ken.fitsInCage(50));
29        assertEquals("Error in FitsInCage()",true,ken.fitsInCage(4));
30    }
31
32 }
33
34
35
```

Failure Trace

```
1 java.lang.AssertionError: Error in isHealthy() expected<true> but was<false>
2 at lab8_202001196_viraj.BoaTest.testIsHealthy(BoaTest.java:20)
```

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
lab8_202001196_viraj	53.9 %	62	53	115

Red line shows that expected output and actual output are not matching.  
There may be faults in code or in our expected output.

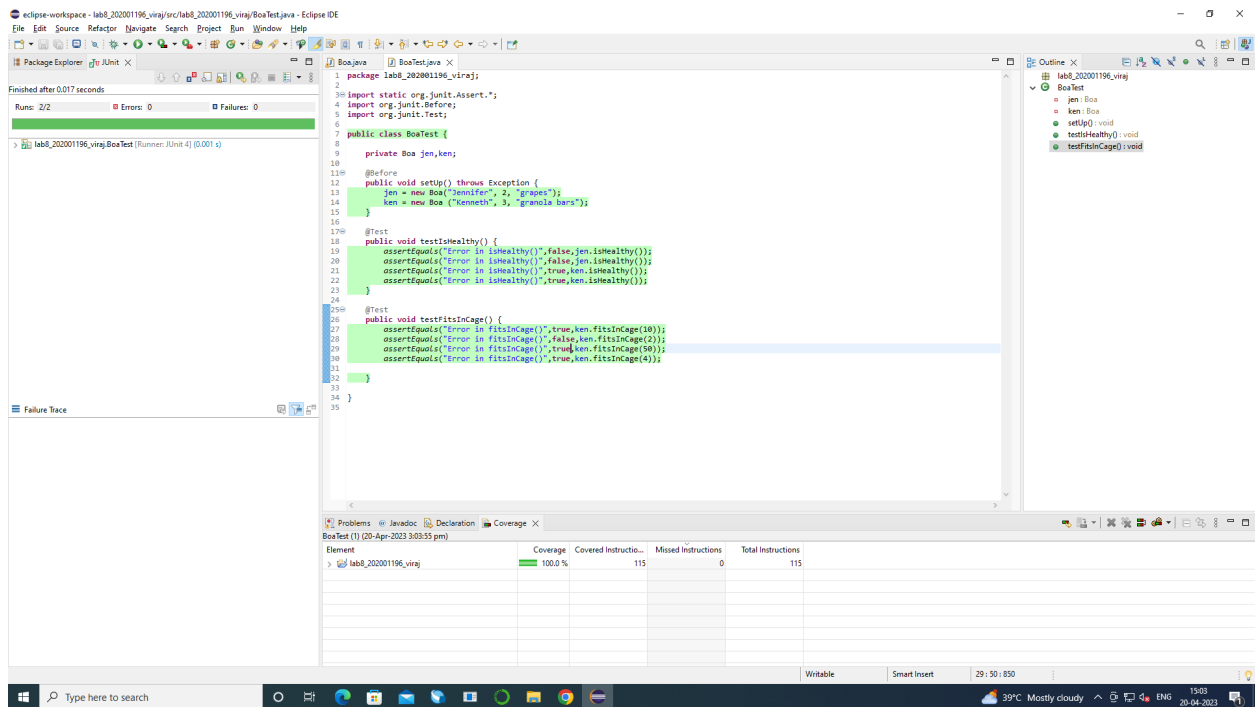
After doing few changes in expected output ,

```
1 package lab8_202001196_viraj;
2
3 import static org.junit.Assert.*;
4 import org.junit.Before;
5 import org.junit.Test;
6
7 public class BoaTest {
8     private Boa jen,ken;
9
10    @Before
11    public void setUp() throws Exception {
12        jen = new Boa("Jennifer", 2, "grapes");
13        ken = new Boa("Kenneth", 3, "granola bars");
14    }
15
16    @Test
17    public void testIsHealthy() {
18        assertEquals("Error in isHealthy()",false,jen.isHealthy());
19        assertEquals("Error in isHealthy()",false,jen.isHealthy());
20        assertEquals("Error in isHealthy()",true,ken.isHealthy());
21        assertEquals("Error in isHealthy()",true,ken.isHealthy());
22    }
23
24    @Test
25    public void testFitsInCage() {
26        assertEquals("Error in FitsInCage()",true,ken.fitsInCage(10));
27        assertEquals("Error in FitsInCage()",false,ken.fitsInCage(2));
28        assertEquals("Error in FitsInCage()",false,ken.fitsInCage(50));
29        assertEquals("Error in FitsInCage()",true,ken.fitsInCage(4));
30    }
31
32 }
33
34
35
```

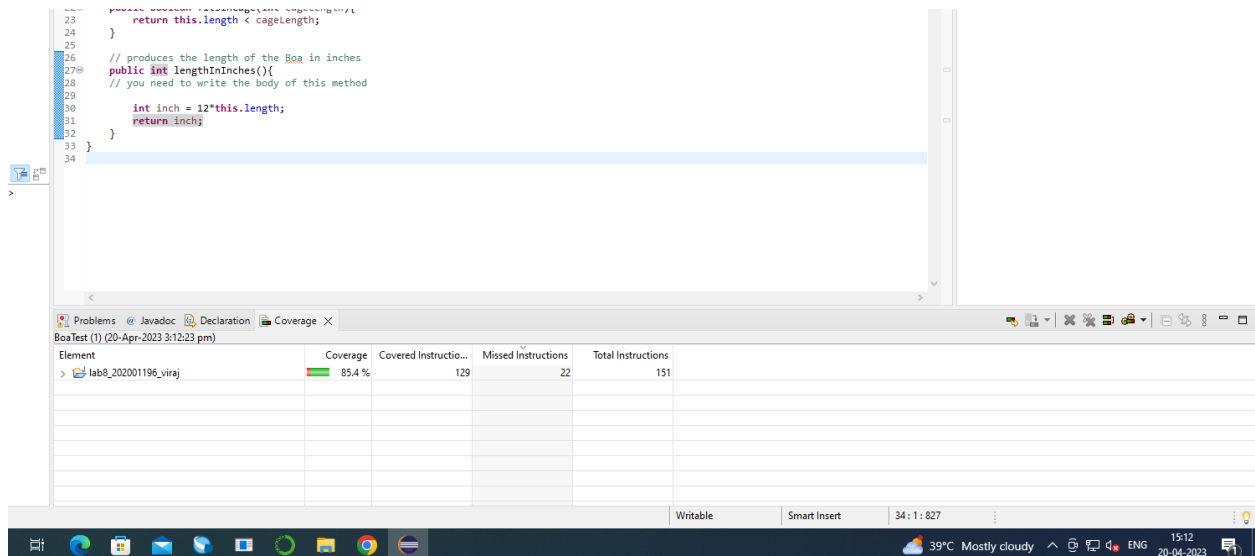
Failure Trace

```
1 java.lang.AssertionError: Error in fitsInCage() expected<false> but was<true>
2 at lab8_202001196_viraj.BoaTest.testFitsInCage(BoaTest.java:29)
```

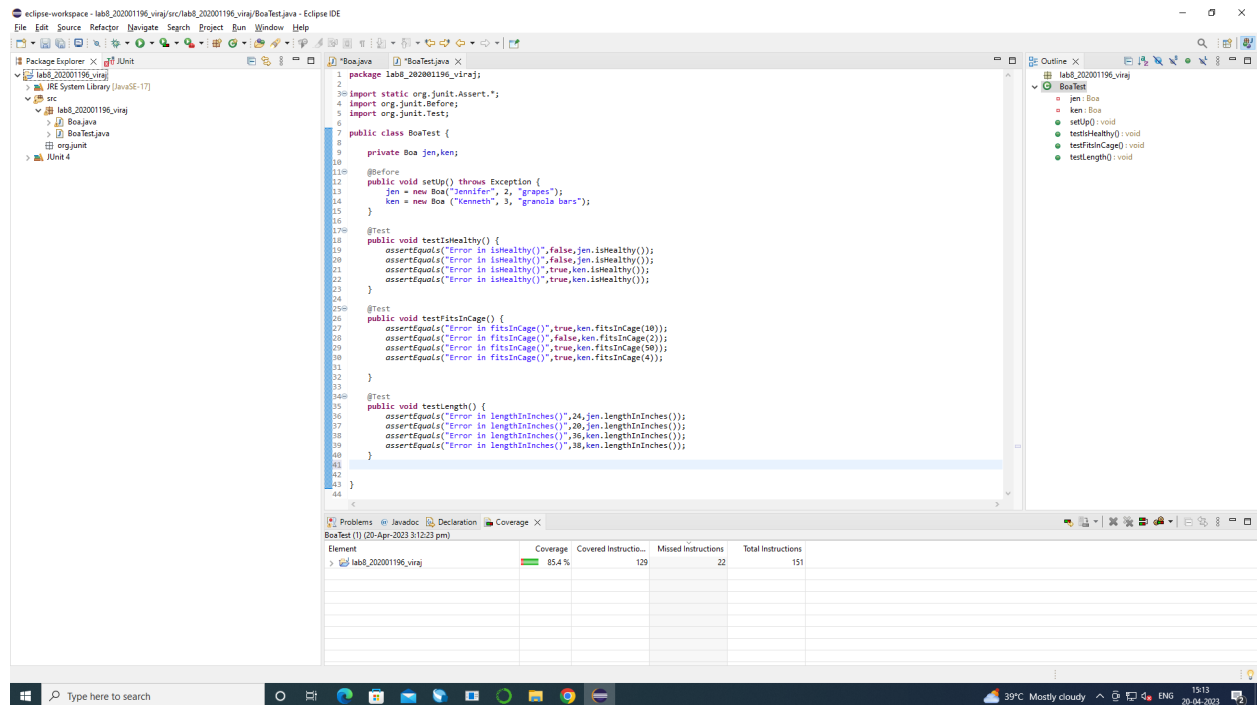
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
lab8_202001196_viraj	83.5 %	96	19	115



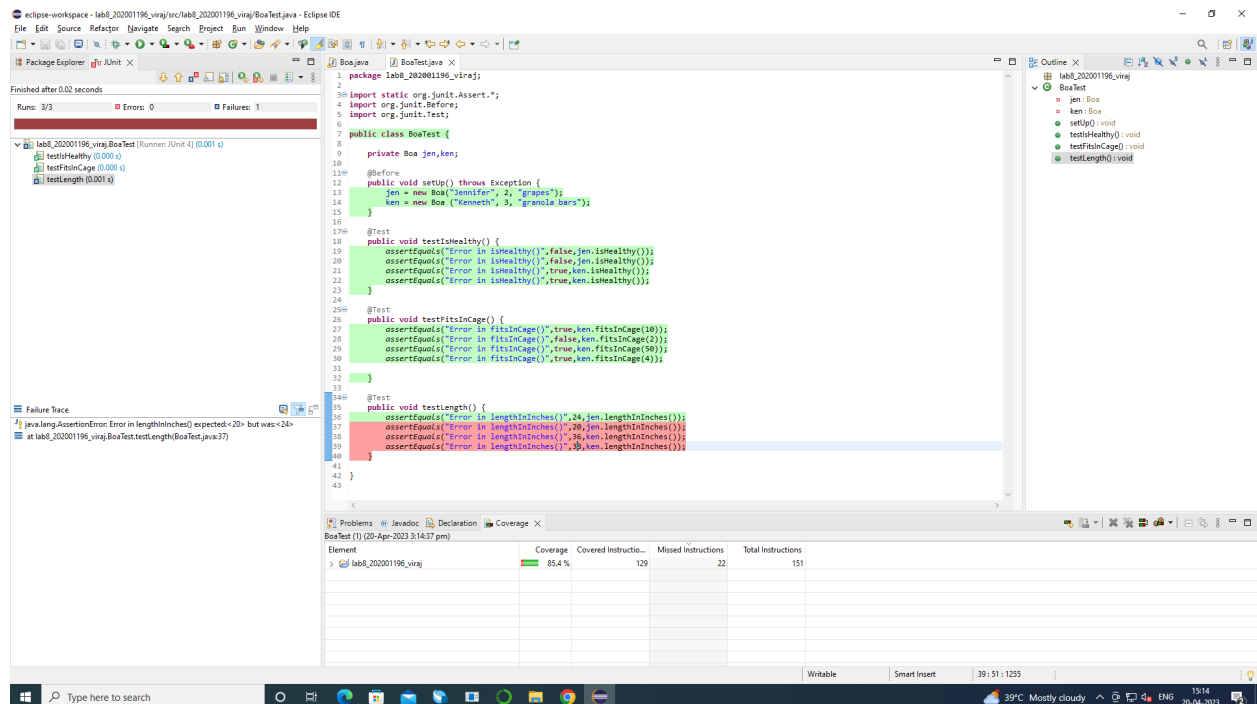
## Adding new function to class :



## Testing file:



## After running:



Same as previous , have to change either expected output OR code,

After changes in expected output ,

The screenshot shows the Eclipse IDE with a Java test class named `BoaTest` in the package `lab8_202001196_viraj`. The class contains several test methods using JUnit assertions. The IDE shows the code editor, a console window, and a coverage report.

```
1 package lab8_202001196_viraj;
2
3 import static org.junit.Assert.*;
4 import org.junit.Before;
5 import org.junit.Test;
6
7 public class BoaTest {
8
9     private Boa jen, ken;
10
11     @Before
12     public void setUp() throws Exception {
13         jen = new Boa("Jennifer", 2, "grapes");
14         ken = new Boa("Kenneth", 3, "granola bars");
15     }
16
17     @Test
18     public void testIsHealthy() {
19         assertEquals("Error in isHealthy()", false, jen.isHealthy());
20         assertEquals("Error in isHealthy()", false, ken.isHealthy());
21         assertEquals("Error in isHealthy()", true, ken.isHealthy());
22         assertEquals("Error in isHealthy()", true, ken.isHealthy());
23     }
24
25     @Test
26     public void testFitsInCage() {
27         assertEquals("Error in fitsInCage()", true, ken.fitsInCage(10));
28         assertEquals("Error in fitsInCage()", false, ken.fitsInCage(2));
29         assertEquals("Error in fitsInCage()", true, ken.fitsInCage(50));
30         assertEquals("Error in fitsInCage()", true, ken.fitsInCage(4));
31     }
32
33
34     @Test
35     public void testLength() {
36         assertEquals("Error in lengthInches()", 24, jen.lengthInches());
37         assertEquals("Error in lengthInches()", 24, jen.lengthInches());
38         assertEquals("Error in lengthInches()", 36, ken.lengthInches());
39         assertEquals("Error in lengthInches()", 36, ken.lengthInches());
40     }
41
42 }
43
```

The console window shows the test results: "Finished after 0.016 seconds", "Runs: 3/3", "Errors: 0", "Failures: 0". The coverage report at the bottom shows 100.0% coverage for the `BoaTest` class.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
BoaTest (1) (20-Apr-2023 3:16:08 pm)	100.0 %	151	0	151

## Code of Boa Class File:

```
package lab8_202001196_viraj;
public class Boa {

    private String name;
    private int length; // the length of the boa, in feet
    private String favoriteFood;

    public Boa (String name, int length, String favoriteFood){
        this.name = name;
        this.length = length;
        this.favoriteFood = favoriteFood;
    }

    // returns true if this boa constructor is healthy
    public boolean isHealthy(){
        return this.favoriteFood.equals("granola bars");
    }

    // returns true if the length of this boa constructor is
    // less than the given cage length
    public boolean fitsInCage(int cageLength){
        return this.length < cageLength;
    }

    // produces the length of the Boa in inches
    public int lengthInInches(){
        // you need to write the body of this method

        int inch = 12*this.length;
        return inch;
    }
}
```

## Code of Testing File:

```
package lab8_202001196_viraj;
import static org.junit.Assert.*;
import org.junit.Before;
import org.junit.Test;
public class BoaTest {

    private Boa jen,ken;

    @Before
    public void setUp() throws Exception {
```

```

        jen = new Boa("Jennifer", 2, "grapes");
        ken = new Boa ("Kenneth", 3, "granola bars");
    }

    @Test
    public void testIsHealthy() {
        assertEquals("Error in isHealthy()", false, jen.isHealthy());
        assertEquals("Error in isHealthy()", false, jen.isHealthy());
        assertEquals("Error in isHealthy()", true, ken.isHealthy());
        assertEquals("Error in isHealthy()", true, ken.isHealthy());
    }

    @Test
    public void testFitsInCage() {
        assertEquals("Error in fitsInCage()", true, ken.fitsInCage(10));
        assertEquals("Error in fitsInCage()", false, ken.fitsInCage(2));
        assertEquals("Error in fitsInCage()", true, ken.fitsInCage(50));
        assertEquals("Error in fitsInCage()", true, ken.fitsInCage(4));
    }

    @Test
    public void testLength() {
        assertEquals("Error in lengthInInches()", 24, jen.lengthInInches());
        assertEquals("Error in lengthInInches()", 24, jen.lengthInInches());
        assertEquals("Error in lengthInInches()", 36, ken.lengthInInches());
        assertEquals("Error in lengthInInches()", 36, ken.lengthInInches());
    }
}

```