

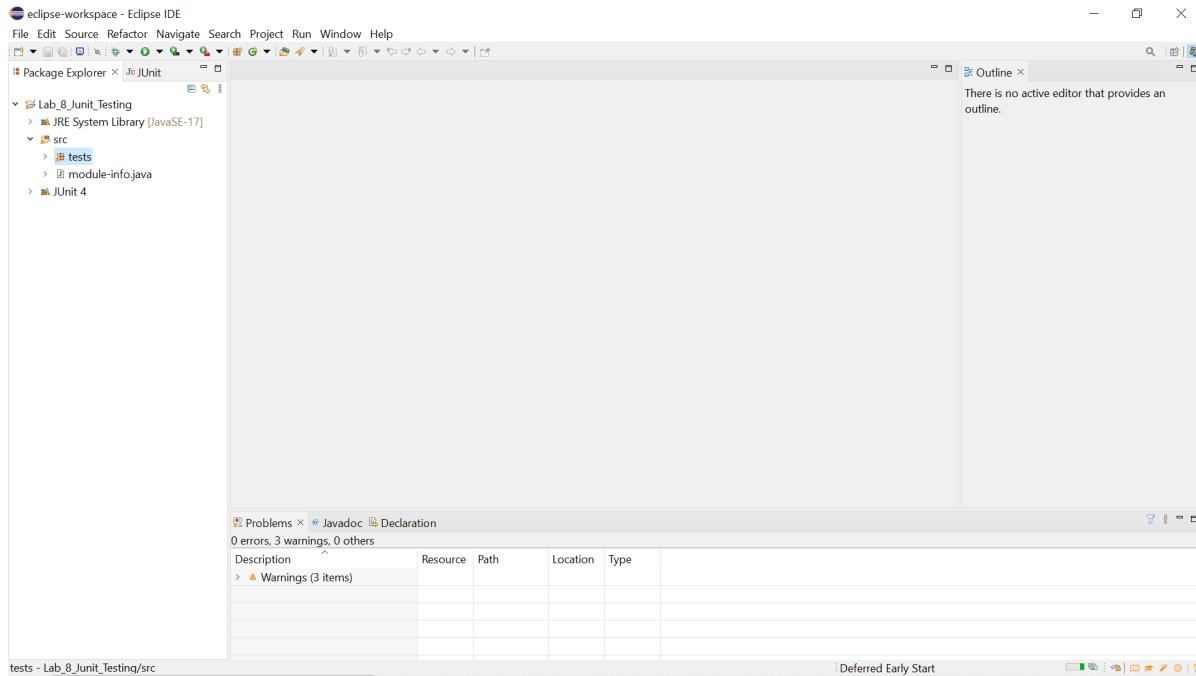
# IT 314 Software Engineering

## Lab 8

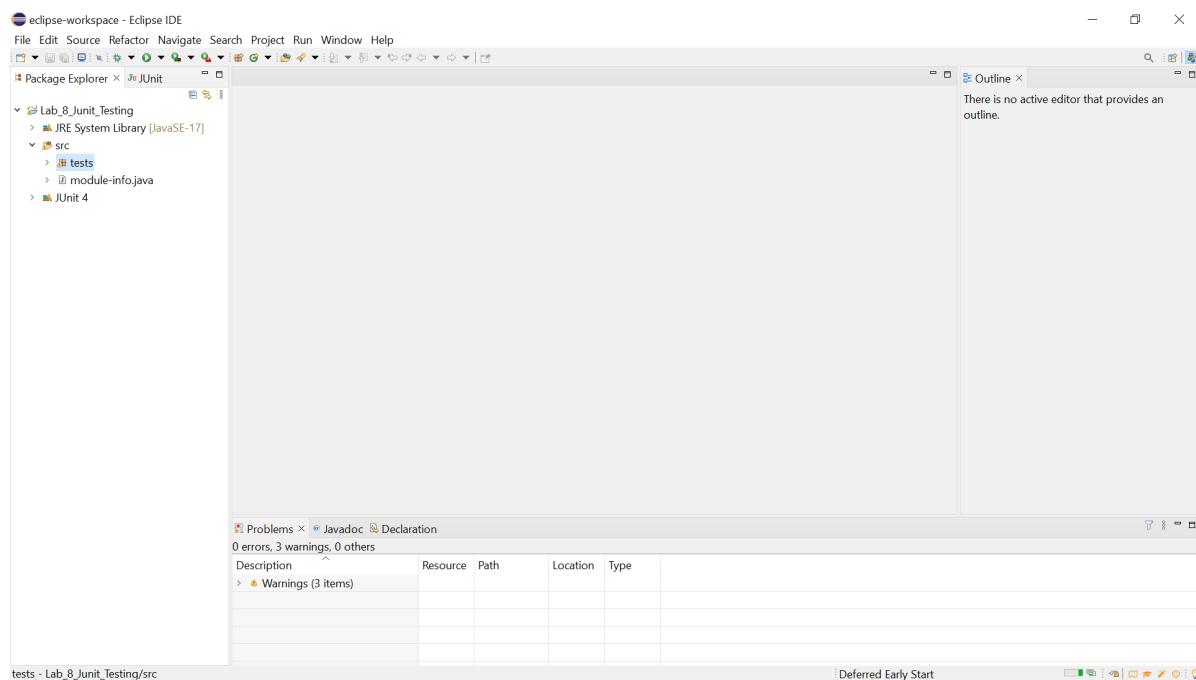
Name: Poriya Neel Dharmeshbhai

ID: 202001139

Create a new Eclipse project,



Create a new class named “Boa” with the following code,



Create a JUnit test case file named “BoaTest”,

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project "Lab\_8\_Junit\_Testing" with a "src" folder containing "tests", "Boajava", and "BoaTest.java".
- Code Editor:** Displays the content of BoaTest.java:

```
1 package tests;
2
3 import static org.junit.Assert.*;
4
5 public class BoaTest {
6
7     @Test
8     public void test() {
9         fail("Test not implemented");
10    }
11
12 }
13
14
15 }
16
```
- Outline View:** Shows the class structure: tests > BoaTest > test(): void.
- Problems View:** Shows 0 errors, 2 warnings, and 0 others. The warnings list contains "Warnings (2 items)".

Create a setup function which initializes two new instances of Bos class named “jen” and “ken”,

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project "Lab\_8\_Junit\_Testing" with a "src" folder containing "tests", "Boajava", and "BoaTest.java".
- Code Editor:** Displays the content of BoaTest.java:

```
1 package tests;
2
3 import static org.junit.Assert.*;
4
5 public class BoaTest {
6
7     @Before
8     public void setUp() throws Exception {
9         jen = new Boa("Jennifer", 2, "grapes");
10        ken = new Boa("Kenneth", 3, "granola bars");
11    }
12
13    @Test
14    public void test() {
15        fail("Test not implemented");
16    }
17
18 }
19
```
- Outline View:** Shows the class structure: tests > BoaTest > jen: Boa > ken: Boa > setUp(): void > test(): void.
- Problems View:** Shows 0 errors, 1 warning, and 0 others. The warning list contains "Warnings (1 item)".

Now we have to implement the testIsHealthy() and testFitsInCage() functions in the “BoaTest” class.

```

eclipse-workspace - Lab_8_JUnit_Testing/src/tests/BoaTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit Boa.java BoaTest.java
10 Boa jen, ken;
11
12 @Before
13 public void setUp() throws Exception {
14     // Initialization
15     jen = new Boa("Jennifer", 2, "grapes");
16     ken = new Boa ("Kenneth", 3, "granola bars");
17 }
18
19 @Test
20 public void testIsHealthy() {
21     //testing isHealthy for two objects
22     assertEquals(false, jen.isHealthy());
23     assertEquals(true, ken.isHealthy());
24 }
25
26 @Test
27 public void testFitsInCage() {
28     // less than cage size
29     assertEquals(false, jen.fitsInCage(1));
30
31     // equal to cage size
32     assertEquals(false, jen.fitsInCage(2));
33
34     // greater than cage size
35     assertEquals(true, jen.fitsInCage(3));
36 }

```

Problems X Javadoc Declaration  
0 errors, 1 warning, 0 others

Description	Resource	Path	Location	Type
> Warnings (1 item)				

Writable Smart Insert 28 : 31 : 554

It is not necessary to develop tests for both ken and jen objects in order to test the fitsInCage() method because the function is the same for both, and the results of test cases depend only on whether the specified length is greater than, less than, or equal to the actual length of the object. In both situations, the behavior will be comparable.

## Running the testcases,

eclipse-workspace - Lab\_8\_JUnit\_Testing/src/tests/BoaTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit Boa.java BoaTest.java

Finished after 0.011 seconds

Runs: 2/2 Errors: 0 Failures: 0

tests.BoaTest [Runner: JUnit 4] (0.001 s)

- testIsHealthy (0.001 s)
- testFitsInCage (0.000 s)

Failure Trace

BoaTest.java

```

13 public void setUp() throws Exception {
14     // Initialization
15     jen = new Boa("Jennifer", 2, "grapes");
16     ken = new Boa ("Kenneth", 3, "granola bars");
17 }
18
19 @Test
20 public void testIsHealthy() {
21     //testing isHealthy for two objects
22     assertEquals(false, jen.isHealthy());
23     assertEquals(true, ken.isHealthy());
24 }
25
26 @Test
27 public void testFitsInCage() {
28     // less than cage size
29     assertEquals(false, jen.fitsInCage(1));
30
31     // equal to cage size
32     assertEquals(false, jen.fitsInCage(2));
33
34     // greater than cage size
35     assertEquals(true, jen.fitsInCage(3));
36 }
37
38 }
39

```

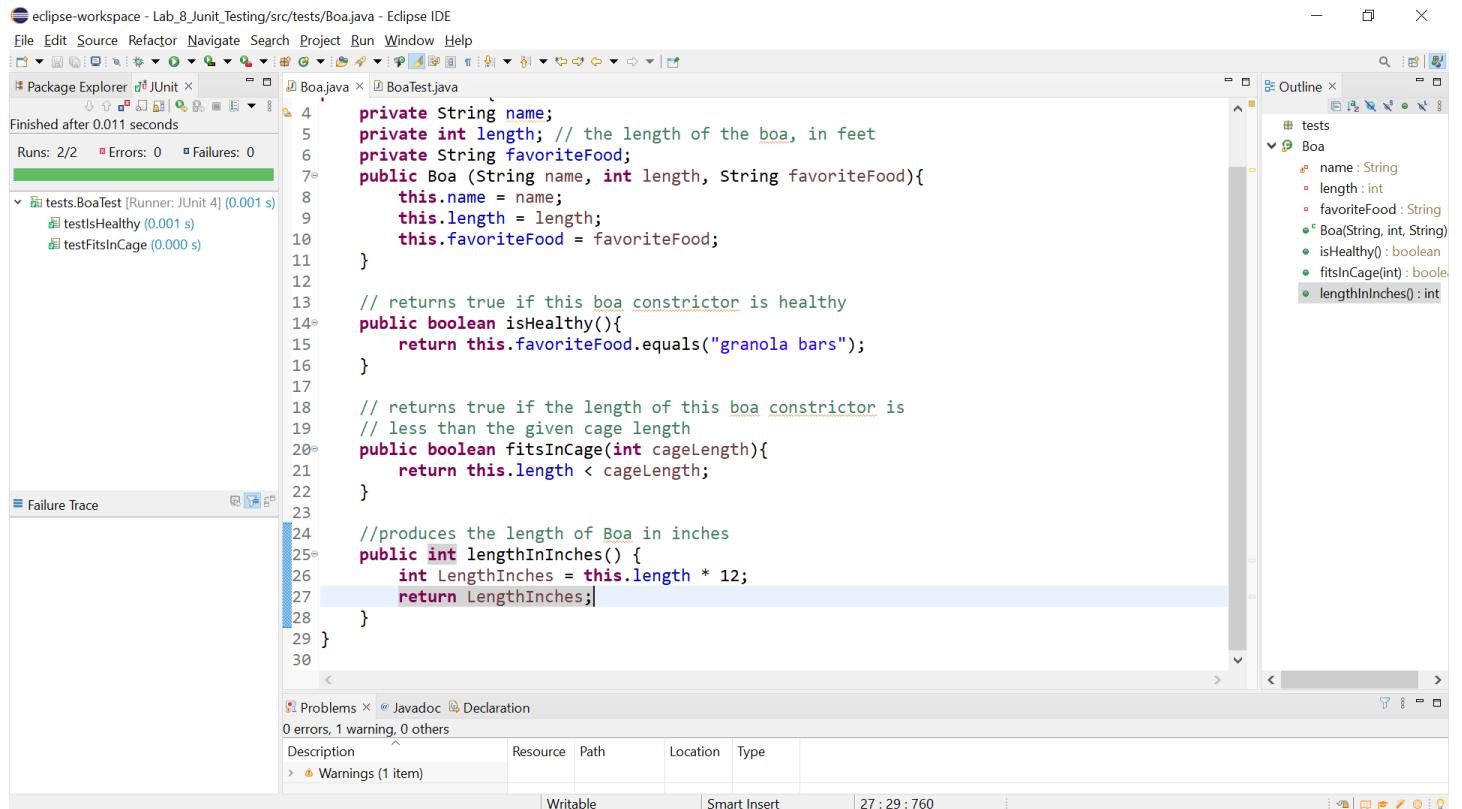
Problems X Javadoc Declaration  
0 errors, 1 warning, 0 others

Description	Resource	Path	Location	Type
> Warnings (1 item)				

Writable Smart Insert 21 : 44 : 399

We can observe that both the tests ran successfully.

Now, we create a new method to the Boa class with name lengthInInches() to get the length in inches.



```
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;
}

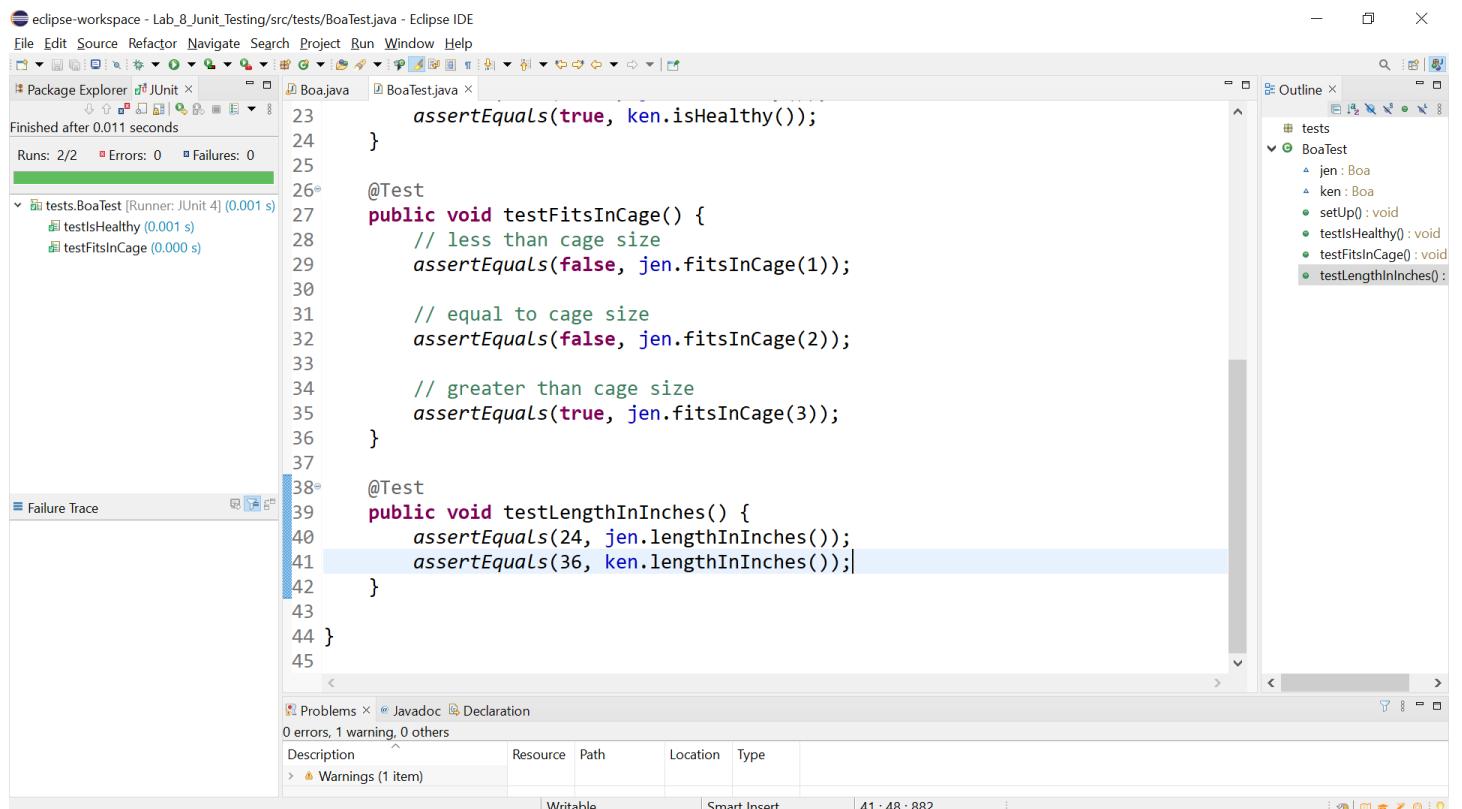
// returns true if this boa constrictor is healthy
public boolean isHealthy(){
    return this.favoriteFood.equals("granola bars");
}

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

//produces the length of Boa in inches
public int lengthInInches() {
    int LengthInches = this.length * 12;
    return LengthInches;
}
```

The Boa's length is specified in feet, so I multiplied length by 12 to convert it to inches and then returned the result.

Now, writing a new test case for testing the length in inches,



```
assertEquals(true, ken.isHealthy());
}

@Test
public void testFitsInCage() {
    // less than cage size
    assertEquals(false, jen.fitsInCage(1));

    // equal to cage size
    assertEquals(false, jen.fitsInCage(2));

    // greater than cage size
    assertEquals(true, jen.fitsInCage(3));
}

@Test
public void testLengthInInches() {
    assertEquals(24, jen.lengthInInches());
    assertEquals(36, ken.lengthInInches());
}
```

Running the newly created test cases,

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** eclipse-workspace - Lab\_8\_Junit\_Testing/src/tests/BoaTest.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard toolbar items like New, Open, Save, etc.
- Package Explorer:** Shows the project structure with 'tests' and 'BoaTest' selected.
- Outline View:** Shows the class hierarchy: tests > BoaTest > jen:Boa > ken:Boa.
- Editor:** Displays the code for BoaTest.java. It includes three test methods: testLengthInInches, testIsHealthy, and testFitsInCage. The last method is currently selected.
- Run View:** Shows the results of the run: Finished after 0.012 seconds, Runs: 3/3, Errors: 0, Failures: 0.
- Failure Trace:** No errors or failures listed.
- Problems View:** Shows 0 errors, 1 warning, 0 others. The warning is 'Warnings (1 item)'.

As a result, test cases have been created for the specified Boa class, and the necessary Junit test cases have been used to test all three methods.

**\*\*\*END OF ASSIGNMENT\*\*\***