

# **Software Engineering IT314**

## Lab-8

Name: Meet Patel

**Student ID: 202001074** 

**Date:** 19/04/2023

### → Below is the provided code for the 'Boa' class created for testing.

```
// represents a boa constrictor

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 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;
}
}</pre>
```

- → Now we create a Junit Test case for testing the above 'Boa' class and named it 'Boa\_test'.
- → Below is the Test code for running the **initial test cases**.

```
package test_lab8;
import static org.junit.jupiter.api.Assertions.*;
import org.junit.Before;
import org.junit.jupiter.api.Test;
class Boa_test {
     @Test
    public void testIsHealthyWithFavoriteFoodGranolaBars() {
        Boa boa = new Boa("Benny", 5, "granola bars");
        assertTrue(boa.isHealthy());
}
@Test
public void testIsHealthyWithFavoriteFoodNotGranolaBars() {
        Boa boa = new Boa("Benny", 5, "mice");
        assertFalse(boa.isHealthy());
}
```

```
public void testFitsInCageWhenLengthLessThanCageLength() {
    Boa boa = new Boa("Benny", 5, "granola bars");
    assertTrue(boa.fitsInCage(10));
  @Test
 public void testFitsInCageWhenLengthGreaterThanCageLength() {
    Boa boa = new Boa("Benny", 20, "granola bars");
    assertFalse(boa.fitsInCage(10));
   → Now we ran the above test cases successfully.
   → Now we use the @Before tag to run another test case.
   → The @Before tag runs before every test case(@Test).
   → The code snippet below show how to execute it.
  private Boa jen;
  private Boa ken;
  @Before
 public void setUp() throws Exception {
       jen = new Boa("Jennifer", 2, "grapes");
       ken = new Boa("Kenneth", 3, "granola bars");
}
```

- → Now performing Q5
- → Below is the test code for the question.

```
@Test
public void testIsHealthy() {
     Boa jen = new Boa("Jen", 5, "granola bars");
   Boa ken = new Boa("Ken", 6, "mice");
  assertTrue(jen.isHealthy());
  assertFalse(ken.isHealthy());
@Test
public void testFitsInCage() {
     Boa jen = new Boa("Jen", 5, "granola bars");
   Boa ken = new Boa("Ken", 6, "mice");
  assertFalse(jen.fitsInCage(2));
  assertTrue(jen.fitsInCage(10));
  assertTrue(jen.fitsInCage(15));
  assertFalse(ken.fitsInCage(4));
  assertTrue(ken.fitsInCage(15));
  assertTrue(ken.fitsInCage(20));
```

→ Now creating another method in the 'Boa' class for Q7.

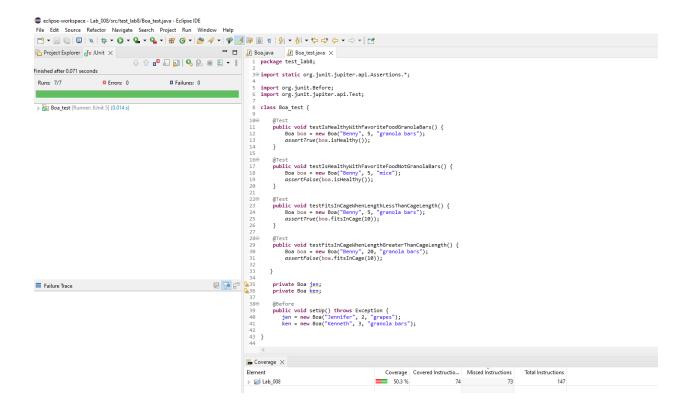
#### **Method Code:**

```
public int lengthInInches() {
  return this.length * 12;
}
```

## **Testing code:**

```
@Test
  public void testLengthInInches() {
    Boa boa = new Boa("John", 5, "grapes");
    int expectedLengthInInches = 60;
    int actualLengthInInches = boa.lengthInInches();
    assertEquals(expectedLengthInInches, actualLengthInInches);
}
```

→ Below is the snippet for the complete testing code.



```
Project Explorer Tu JUnit X
                                  Finished after 0.071 seconds
Runs: 7/7 ■ Errors: 0
                                                                                  @Test
public void testlengthInInches() {
   Boa boa = new Boa("John", 5, "grapes");
   int expectedLengthInInches = 60;
   int actualLengthInInches = boa.lengthInInches();
   ossertEquols(expectedLengthInInches, actualLengthInInches);
}
 > Boa_test [Runner: JUnit 5] (0.014 s)
                                                                                            assertTrue(jen.isHealthy());
assertFalse(ken.isHealthy());
}
                                                                                            @Test
public void testFitsInCage() {
    Boa jen = new Boa("Jen", 5, "granola bars");
    "" " c "mice");
                                                                                                  Boa ken = new Boa("Ken", 6, "mice");
                                                                                               assertFalse(jen.fitsInCage(2));
assertTrue(jen.fitsInCage(10));
assertTrue(jen.fitsInCage(15));
assertFalse(ken.fitsInCage(4));
assertTrue(ken.fitsInCage(15));
assertTrue(ken.fitsInCage(20));
                                                                     B 7 #
Failure Trace
                                                                                  Coverage ★
                                                                                                                                     Coverage Covered Instruction... Missed Instructions Total Instructions 50.3 % 74 73 147
                                                                                  Element
                                                                                   > 📂 Lab_008
```