

IT314 -Software Engineering

Lab-8 Unit Testing with JUnit

Name: Aniruddh Damor

ID: 202001255

➤ **Sample 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;
    }
}
```

➤ **Now we create a Junit Test case for testing the above ‘Boa’ class and named it ‘Boa_test’.**

```
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());
    }

    @Test
    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).

➤ **Q4: 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");
}

```

➤ **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));
}

```

➤ **Q7: Creating another method in the 'Boa' class.**

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);  
}
```

- Here is the snippet for the complete testing code.

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

- Project Explorer:** Shows the project structure with 'Boa_test' selected.
- JUnit Runner:** Displays the test results: 'Finished after 0.071 seconds', 'Runs: 7/7', 'Errors: 0', and 'Failures: 0'.
- Code Editor:** Shows the source code for 'Boa_test.java'. The code includes imports for JUnit, a class definition for 'Boa_test', and several test methods: 'testIsHealthyWithFavoriteFoodGranolaBars()', 'testIsHealthyWithFavoriteFoodNotGranolaBars()', 'testFitsInCageWhenLengthLessThanCageLength()', and 'testFitsInCageWhenLengthGreaterThanCageLength()'. It also includes a 'setUp()' method for initializing test data.
- Coverage Report:** A table at the bottom showing the coverage for the 'Lab_008' element.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Lab_008	50.3 %	74	73	147

File Edit Source Refactor Navigate Search Project Run Window Help

Project ExplorerJUnit X

Boa.javaBoa_test.java X

38@
39 @Before
40 public void setUp() throws Exception {
41 jen = new Boa("Jennifer", 2, "grapes");
42 ken = new Boa("Kenneth", 3, "granola bars");
43 }
44
45 @Test
46 public void testLengthInInches() {
47 Boa boa = new Boa("John", 5, "grapes");
48 int expectedLengthInInches = 60;
49 int actualLengthInInches = boa.lengthInInches();
50 assertEquals(expectedLengthInInches, actualLengthInInches);
51 }
52
53 @Test
54 public void testIsHealthy() {
55 Boa jen = new Boa("Jen", 5, "granola bars");
56
57 Boa ken = new Boa("Ken", 6, "mice");
58
59 assertTrue(jen.isHealthy());
60 assertFalse(ken.isHealthy());
61 }
62
63 @Test
64 public void testFitsInCage() {
65 Boa jen = new Boa("Jen", 5, "granola bars");
66
67 Boa ken = new Boa("Ken", 6, "mice");
68
69 assertFalse(jen.fitsInCage(2));
70 assertTrue(jen.fitsInCage(10));
71 assertTrue(jen.fitsInCage(15));
72 assertFalse(ken.fitsInCage(4));
73 assertTrue(ken.fitsInCage(15));
74 assertTrue(ken.fitsInCage(20));
75 }
76
77
78
79
80
81

Finished after 0.071 seconds
Runs: 7/7Errors: 0Failures: 0
Boa_test [Runner: JUnit 5] (0.014 s)

Failure Trace

Coverage X

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
> Lab_008	50.3 %	74	73	147