

In this lab we will check out some java code from a github repository or you can create your own code, then perform some white-box testing on it. We will use a tool called EMMA as a plugin to Eclipse to measure the coverage of code. In addition to that we will also use FindBugs for static code analysis

Attendance/Demo

To receive credit for this lab, you must make reasonable progress towards completing the exercises. When you have finished all the exercises, call your TA, who will review your work. For those who don't finish early, the TA will ask you to show whatever diagrams you have completed, starting at about 15 minutes before the end of the lab period. Finish any exercises that you don't complete on your own time.

Steps

Installing EMMA

The Emma based plugin for Eclipse is ECLemma. To install:

1. From Eclipse Menu select Help →Eclipse Marketplace.
2. Search for EclEmma
3. Hit Install for the Entry "EclEmma Java Code Coverage".
4. Follow the steps and install.
5. Wait... before you restart Eclipse

Installing FindBugs

1. From Eclipse Menu select Help →Eclipse Marketplace.
2. Search for FindBugs or go to add and copy paste <http://findbugs.cs.umd.edu/eclipse-candidate> as the location.
3. Follow the steps and install.
4. Restart Eclipse

Executing Code

Now run your code in eclipse. You can follow either of the options

Run your project's code in eclipse

Or

Use the project from the Git repository

Or

Use the following code

The code has a class named person with getter and setter methods and a constructors.

This is a simple piece of code to give you a demonstration of the junit tests and using

Emma Coverage tool.

```
package or.packag; //Create and select a package for creating the class.

public class person {

    //Attributes
    private String name;
    private int number;

    //constructors
    public person()
    {
        name="XYZ";
        number=3;
    }

    //getters and setters
    public String getName()
    {
        return name;
    }
    public void setName(String anyName) {
        name=anyName;
    }
    public int getNumber() {
        return number;
    }
    public void setNumber(int number) {
        this.number = number;
    }
}
```

Creating JUnit Tests

To create JUnit tests for this class:

Select the project in package explorer →right click →new→JUnit test case

Name the test case, at this point you can also name the class you want to test e.g. person.

The Code for the junit will look something like this.

```
package or.testpack;
```

```
import or.packag.person;
```

```
import junit.framework.TestCase;
```

```
public class personTest extends TestCase {    //personTest extends the TestCase Class
                                              for testing

    public void testPerson() {
```

```

    person p1= new person();          //Creating the object of the class to be tested
    assertEquals("XYZ", p1.getName()); // Since this will result in testing the constructor
                                        XYZ here corresponds to the name in the
                                        constructor. Try changing the name and check
                                        the coverage using EMMA (Discussion coming
                                        next..)

    assertEquals(3, p1.getNumber());
}

public void testSetName() {           //Testing SetName Method
    person p2 = new person();
    p2.setName("EMMA");
    assertEquals("EMMA", p2.getName());
}

public void testSetNumber() {         //Testing SetNumber Method
    person p3= new person();
    p3.setNumber(12);
    assertEquals(12, p3.getNumber());
}

}

```

Observing Coverage

After a run you can view the coverage directly in eclipse.

Select Run→Coverage Last Launch

The executed lines are highlighted in green, the errors are highlighted in red and the conditional statements are in yellow.

You can also see the coverage values summarized in a table with the help of a coverage view.

Window→Show view→Java→coverage.

To Run FindBugs:

Right click on java project in package explorer→select FindBugs

To try other bugs, check out a random java project on github or use your own code.