**Nitin Nagavel**

# Assignment #2 Software Testing

# Using an Automated Test Generation Tool

Com S/SE 417 Spring 2022

Handed out Feb 17th, 2022

*Due at 11:59 PM, Feb 24th, as a pdf uploaded to Canvas*

## Homework Policy

**Homework Policy:** The homework assignment should be done individually. You may talk to classmates about the problems in general, and you can help each other with getting the tools running, but you must complete the homework on your own. You are not permitted to use published answers from websites, etc. Assistance by others must be specifically credited in the solution to the problem that is turned in, describing what the contribution was (e.g., "Thanks to [name] for explaining X in Problem Y", not "Thanks to [name] for help with HW1."). The Dean of Students Office offers several good resources on how to avoid plagiarism.

*The goal is for each of you to learn the material sufficiently well to use it productively, to think innovatively, and to develop confidence in your problem-solving abilities.* Feel free to talk to me individually about this if you have any questions.

**Late policy:** *10% penalty per day* (or part of day) for late homework. ***Assignments will not be accepted after March 1st***, unless otherwise arranged/discussed with me.

Some homework problems are adapted from the course textbook, "Introduction to Software Testing", 2nd edition, Ammann & Offutt, 2017. There is a Student Solution Manuals available online with answers to other practice questions https://cs.gmu.edu/~offutt/softwaretest/exer-student.pdf.

---

**Using Evosuite 1.1.0**

https://www.evosuite.org/

I have put a version of Evosuite that can be untarred and used for this exercise. It runs on pyrite. It has the minimal parts of Evosuite (the .jar files and two example programs we will use for this assignment). You are welcome to get the full distribution of Evosuite. It has a maven version and there are two docker versions you can use.

You can run these locally on other machines (if you want to set it up). There are directions on using pyrite. Note: you will need the Ant build system (apache.ant.org), JUnit, and Java on your own machine. Note, you will need Java 1.11 or earlier. You may run into problems with evosuite on later versions of Java. You are welcome to try the Maven version.

If you run this on pyrite you can control your version of java by selecting the directory (you can script this to make it easier)

Replace (java or javac below with):

```
/usr/lib/jvm/java-11/bin/java
/usr/lib/jvm/java-11/bin/javac
```

For each numbered item answer the questions, and provide screen shots where asked. Create a .pdf of the final report and submit via canvas. Make sure to clearly label the question #s. I have put ** ** around the text and bolded parts that need to be handed in.

**There is nothing to submit for #1.**

1. Download the evosuite program from Canvas and untar this (pyrite is a good place for this).

I have given you two programs that you can use for generating test suites

One is a stack class (Tutorial_Stack/Stack.java) and the other is our triangle program:

(Triangle/TriangleType.java and Triangle/Triangle.java)

Read the evosuite tutorial to get a sense of what you will be doing (evosuite.org).

2. Generate test cases for both programs using the following steps

First compile both programs

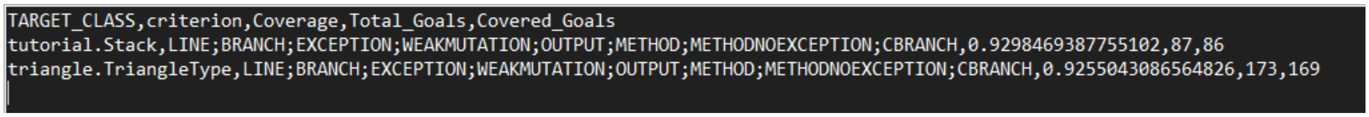
```
> javac Tutorial_Stack/tutorial/*.java
> javac Triangle/triangle/*.java
```

Now run evosuite on each program (note you use the -projectCP flag to tell it which directory and the -class to tell it which class you want to generate tests for)

```
 java -jar evosuite-1.1.0.jar -projectCP Tutorial_Stack -class tutorial.Stack
   java -jar evosuite-1.1.0.jar -projectCP Triangle -class triangle.TriangleType
```

Take a look at the tests (under evosuite-tests) and the reports (under evosuite-reports)

** Copy the contents of the statistics.csv** file to this report



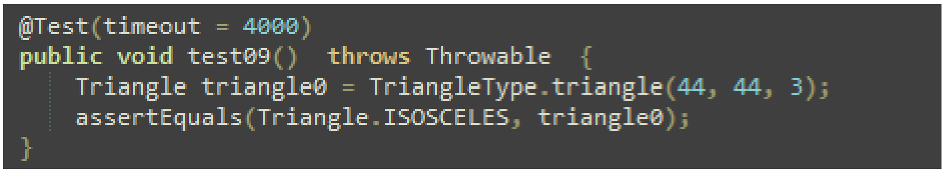
```
TARGET_CLASS,criterion,Coverage,Total_Goals,Covered_Goals
tutorial.Stack,LINE;BRANCH;EXCEPTION;WEAKMUTATION;OUTPUT;METHOD;METHODNOEXCEPTION;CBRANCH,0.9298469387755102,87,86
triangle.TriangleType,LINE;BRANCH;EXCEPTION;WEAKMUTATION;OUTPUT;METHOD;METHODNOEXCEPTION;CBRANCH,0.9255043086564826,173,169
```

Open the test files and answer the following questions:

**a. How many tests are there for each program? **

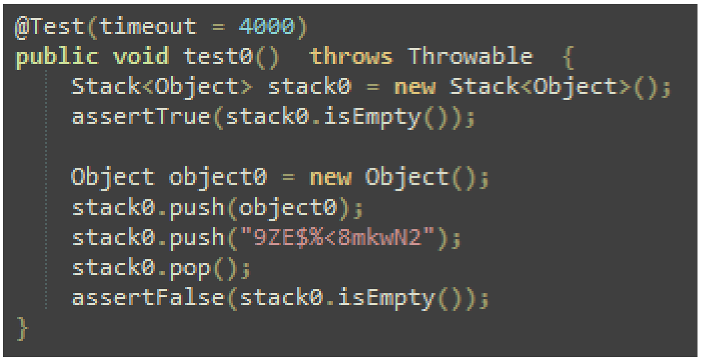
TriangleType – 19 Tests, Tutorial – 6 tests

**b. Show an example of one test for ISOSCELES**



```
@Test(timeout = 4000)
public void test09()  throws Throwable  {
    Triangle triangle0 = TriangleType.triangle(44, 44, 3);
    assertEquals(Triangle.ISOSCELES, triangle0);
}
```

**c. Show an example of the first stack test**



```
@Test(timeout = 4000)
public void test0()  throws Throwable  {
    Stack<Object> stack0 = new Stack<Object>();
    assertTrue(stack0.isEmpty());

    Object object0 = new Object();
    stack0.push(object0);
    stack0.push("9ZE$%<8mkwN2");
    stack0.pop();
    assertFalse(stack0.isEmpty());
}
```

Now export your class path. You need to include the evosuite-tests, the testing jars and the paths

to the programs.

export CLASSPATH=.:evosuite-tests:lib/junit-4.13.jar:lib/hamcrest-core-1.3.jar:lib:Triangle:Tutorial_Stack:evosuite-standalone-runtime-1.1.0.jar

Now compile the tests

```
javac evosuite-tests/triangle/*.java
javac evosuite-tests/tutorial/*.java
```

And run them:

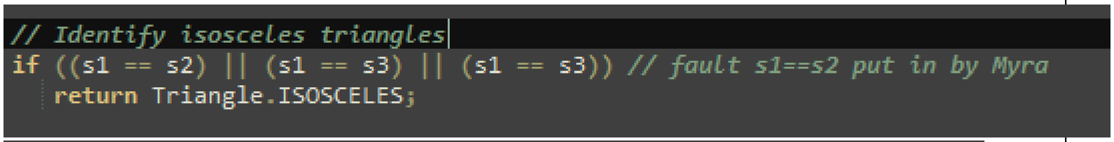
```
java org.junit.runner.JUnitCore tutorial.Stack_ESTest
java org.junit.runner.JUnitCore triangle.TriangleType_EST
```

Answer the following question.

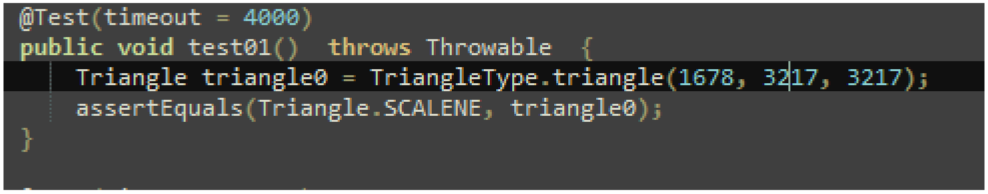
1. **Do both of the test sets pass?**

a Both tests were passing. However in triangle one of the tests, was passing when it should've failed:

b



```
// Identify isosceles triangles
if ((s1 == s2) || (s1 == s3) || (s1 == s3)) // fault s1==s2 put in by Myra
    return Triangle.ISOSCELES;
```

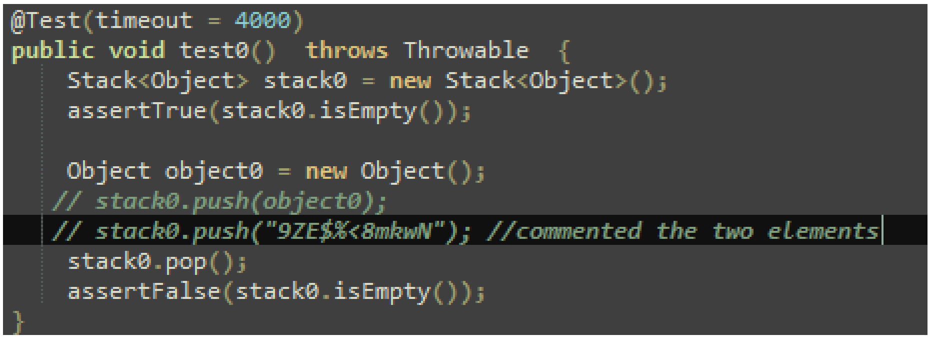


```
@Test(timeout = 4000)
public void test01()  throws Throwable  {
    Triangle triangle0 = TriangleType.triangle(1678, 3217, 3217);
    assertEquals(Triangle.SCALENE, triangle0);
}
```

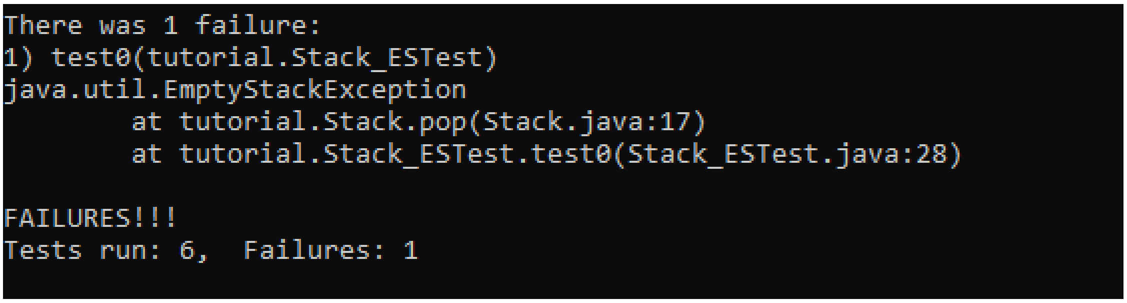
c

2.

**Change the oracle for the first Stack Test and recompile/run the tests. Show the failure as a screenshot**



```
@Test(timeout = 4000)
public void test0()  throws Throwable  {
    Stack<Object> stack0 = new Stack<Object>();
    assertTrue(stack0.isEmpty());

    Object object0 = new Object();
   // stack0.push(object0);
   // stack0.push("9ZE$%<8mkwN"); //commented the two elements
    stack0.pop();
    assertFalse(stack0.isEmpty());
}
```



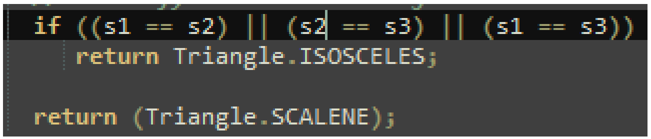
```
There was 1 failure:
1) test0(tutorial.Stack_ESTest)
java.util.EmptyStackException
        at tutorial.Stack.pop(Stack.java:17)
        at tutorial.Stack_ESTest.test0(Stack_ESTest.java:28)

FAILURES!!!
Tests run: 6,  Failures: 1
```

Look more closely at the triangle tests. There is a fault in the program for the isosceles test (see my comment).

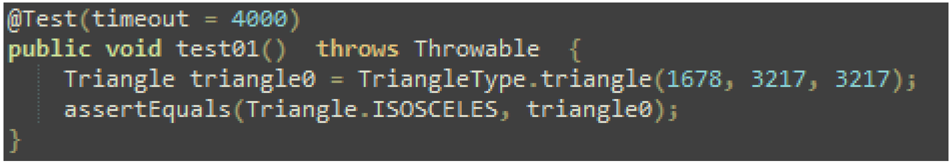
Fix the test case oracle and re-generate evosuite test cases (and make sure they all pass)

Thanks to Adam Banwell for showing me what was going wrong with the ISOSCELES Triangle Test Errors!

3. **Show your correct program**



```
 if ((s1 == s2) || (s2 == s3) || (s1 == s3))
     return Triangle.ISOSCELES;

 return (Triangle.SCALENE);
```

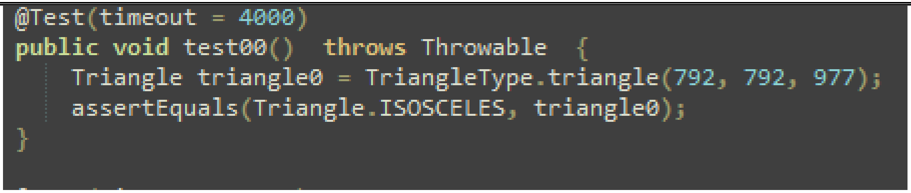


```
@Test(timeout = 4000)
public void test01()  throws Throwable  {
    Triangle triangle0 = TriangleType.triangle(1678, 3217, 3217);
    assertEquals(Triangle.ISOSCELES, triangle0);
}
```

First ScreenShot, I fixed the second condition. The second ScreenShot, It was SCALENE, but I changed it to ISOSCELES

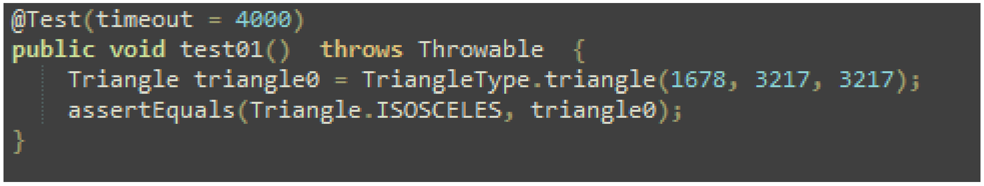
4. **Show the test cases for ISOSCELES triangles**

a



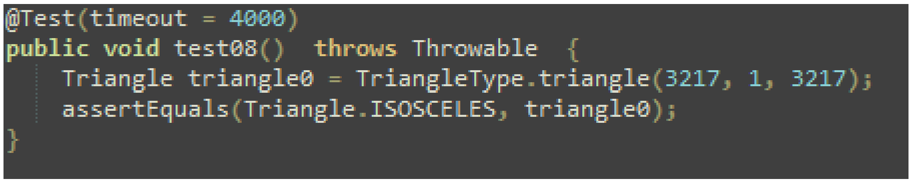
```
@Test(timeout = 4000)
public void test00()  throws Throwable  {
    Triangle triangle0 = TriangleType.triangle(792, 792, 977);
    assertEquals(Triangle.ISOSCELES, triangle0);
}
```

b



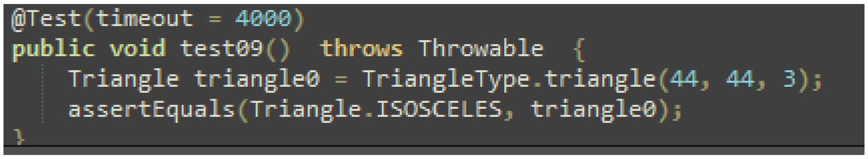
```
@Test(timeout = 4000)
public void test01()  throws Throwable  {
    Triangle triangle0 = TriangleType.triangle(1678, 3217, 3217);
    assertEquals(Triangle.ISOSCELES, triangle0);
}
```

c



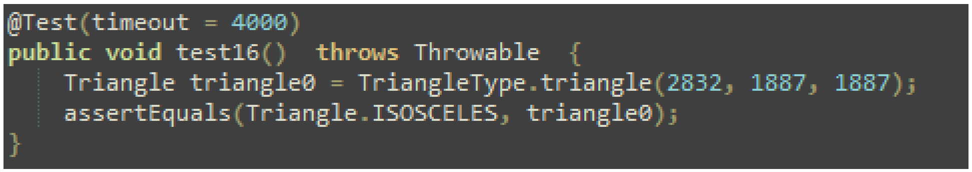
```
@Test(timeout = 4000)
public void test08()  throws Throwable  {
    Triangle triangle0 = TriangleType.triangle(3217, 1, 3217);
    assertEquals(Triangle.ISOSCELES, triangle0);
}
```

d



```
@Test(timeout = 4000)
public void test09()  throws Throwable  {
    Triangle triangle0 = TriangleType.triangle(44, 44, 3);
    assertEquals(Triangle.ISOSCELES, triangle0);
}
```

e

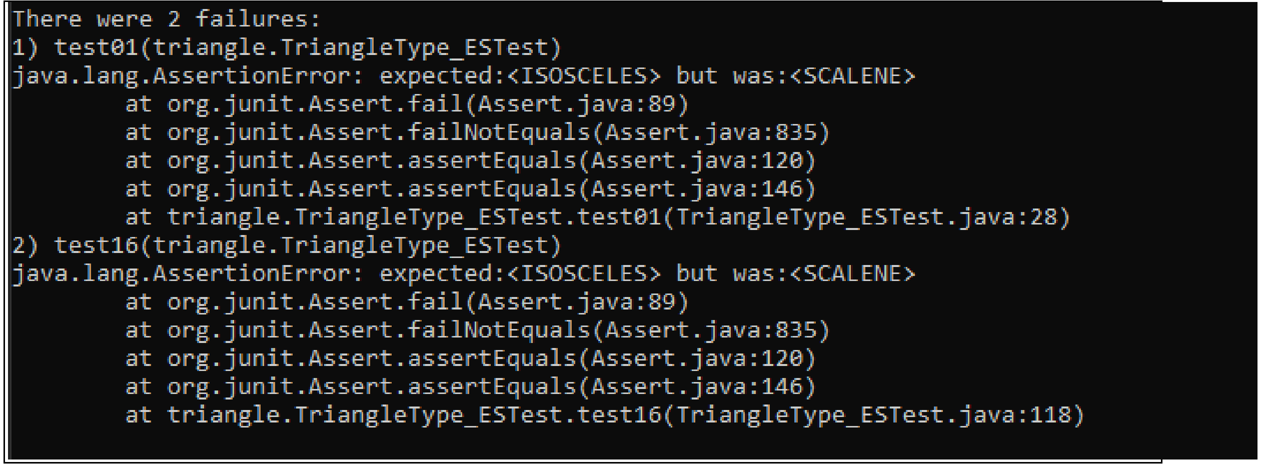


```
@Test(timeout = 4000)
public void test16()  throws Throwable  {
    Triangle triangle0 = TriangleType.triangle(2832, 1887, 1887);
    assertEquals(Triangle.ISOSCELES, triangle0);
}
```

5. **How many test cases do you have now?**  I have 19 Test Cases altogether

Now re-run the test cases using the faulty program version

6. **Provide a screenshot of the result and explain what happened **



```
There were 2 failures:
1) test01(triangle.TriangleType_ESTest)
java.lang.AssertionError: expected:<ISOSCELES> but was:<SCALENE>
        at org.junit.Assert.fail(Assert.java:89)
        at org.junit.Assert.failNotEquals(Assert.java:835)
        at org.junit.Assert.assertEquals(Assert.java:120)
        at org.junit.Assert.assertEquals(Assert.java:146)
        at triangle.TriangleType_ESTest.test01(TriangleType_ESTest.java:28)
2) test16(triangle.TriangleType_ESTest)
java.lang.AssertionError: expected:<ISOSCELES> but was:<SCALENE>
        at org.junit.Assert.fail(Assert.java:89)
        at org.junit.Assert.failNotEquals(Assert.java:835)
        at org.junit.Assert.assertEquals(Assert.java:120)
        at org.junit.Assert.assertEquals(Assert.java:146)
        at triangle.TriangleType_ESTest.test16(TriangleType_ESTest.java:118)
```

a So with the condition commented out, all triangles with s2 == s3, will fall back to the default triangle type of SCALENE

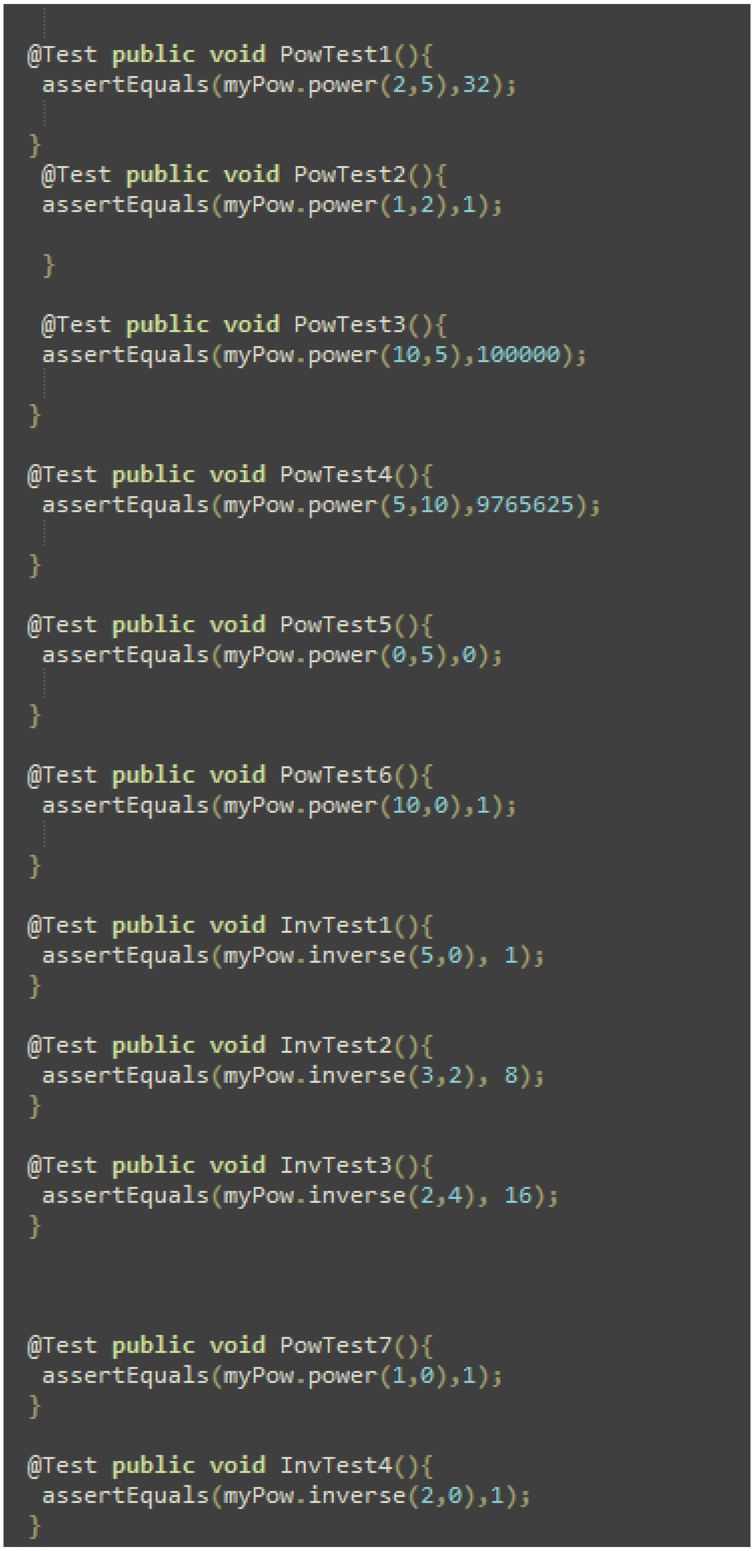
7. **Discuss what you have observed with the faulty test cases/program and the implications for automated test generation**

a With automated test cases, all code logic isn’t tested, but the test cases will be tested the way they are written. By not correctly implementing the  logic, the tests are passing when they should be failing.

Now add your power.java program to the evosuite directory (you can make a directory called “Power” to keep this clean. You may also want to make this a package (i.e. create a directory for the main program and add ‘package power’ in the program). Generate and run tests (**you will have to add “Power” to the classpath to compile/run the Junit Tests**)

8. **How many test cases do you have?**

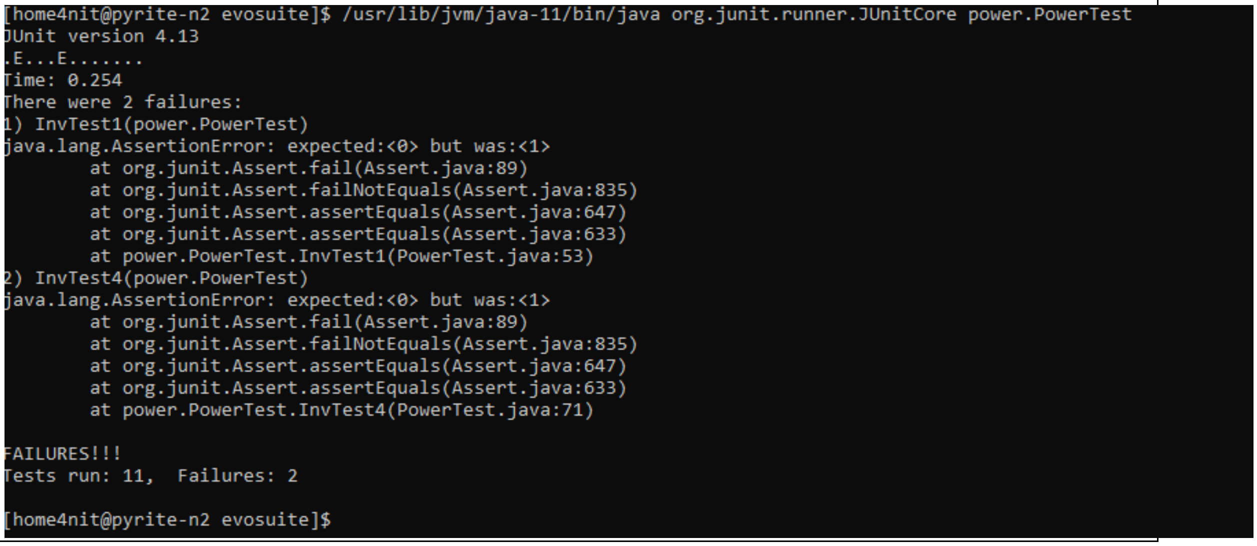
a I have 11 Test Cases



```
@Test public void PowTest1(){
 assertEquals(myPow.power(2,5),32);

}
 @Test public void PowTest2(){
 assertEquals(myPow.power(1,2),1);

 }

 @Test public void PowTest3(){
 assertEquals(myPow.power(10,5),100000);

}

@Test public void PowTest4(){
 assertEquals(myPow.power(5,10),9765625);

}

@Test public void PowTest5(){
 assertEquals(myPow.power(0,5),0);

}

@Test public void PowTest6(){
 assertEquals(myPow.power(10,0),1);

}

@Test public void InvTest1(){
 assertEquals(myPow.inverse(5,0), 1);
}

@Test public void InvTest2(){
 assertEquals(myPow.inverse(3,2), 8);
}

@Test public void InvTest3(){
 assertEquals(myPow.inverse(2,4), 16);
}



@Test public void PowTest7(){
 assertEquals(myPow.power(1,0),1);
}

@Test public void InvTest4(){
 assertEquals(myPow.inverse(2,0),1);
}
```

b

9. **Show the output of running the tests**



```
[home4nit@pyrite-n2 evosuite]$ /usr/lib/jvm/java-11/bin/java org.junit.runner.JUnitCore power.PowerTest
JUnit version 4.13
.E...E.......
Time: 0.254
There were 2 failures:
1) InvTest1(power.PowerTest)
java.lang.AssertionError: expected:<0> but was:<1>
        at org.junit.Assert.fail(Assert.java:89)
        at org.junit.Assert.failNotEquals(Assert.java:835)
        at org.junit.Assert.assertEquals(Assert.java:647)
        at org.junit.Assert.assertEquals(Assert.java:633)
        at power.PowerTest.InvTest1(PowerTest.java:53)
2) InvTest4(power.PowerTest)
java.lang.AssertionError: expected:<0> but was:<1>
        at org.junit.Assert.fail(Assert.java:89)
        at org.junit.Assert.failNotEquals(Assert.java:835)
        at org.junit.Assert.assertEquals(Assert.java:647)
        at org.junit.Assert.assertEquals(Assert.java:633)
        at power.PowerTest.InvTest4(PowerTest.java:71)

FAILURES!!!
Tests run: 11,  Failures: 2

[home4nit@pyrite-n2 evosuite]$
```