

## Assignment #2 Software Testing

Com S/SE 417 Spring 2023

Handed out Feb 14<sup>th</sup>, 2023

*Due at 9 PM, Feb 23<sup>rd</sup>, 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 Feb 28<sup>th</sup>**, unless otherwise arranged/discussed with me.

### Late Policy One Slack Period for all Homework

During the semester you have one "slack period". This means that you can hand in one homework late – up until the late acceptance date – (e.g. Feb 14th for this assignment) without penalty. No questions asked. Once you use up your slack period late penalties will apply as per the late policy

Some homework problems are adapted from the course textbook, "Introduction to Software Testing", 2<sup>nd</sup> 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.

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 unzip/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 a variant of 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
```

**3. 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)

**Handin**

\*\*\* (a) Copy the contents of the statistics.csv\*\*\* file to this report

Open the test files and answer the following questions:

#### Handin

**\*\* (b) How many tests are there for each program? \*\***

**\*\* (c) Show an example of one test for ISOSCELES \*\***

**\*\* (d) Show an example of a stack test that maximizes the stack \*\***

4. **Now export your class path.** You need to include the evosuite-tests, the testing jars and the paths to the programs. (note – you will need to type this in on a single line. All of the libraries are in the “lib:” folder if you want to see their names)

```
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
```

#### 5. 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_ESTest
```

Answer the following question.

**\*\* (a) Look more closely at the triangle tests. There is a fault in the program for the isosceles test.**

What happens with the test generation tool when it creates tests for the isosceles branch? State why this is a limitation of automated test generation tools.

5. **Now copy the “power.java”** program from our last assignment. Make it a package (see class discussion) and put into the evosuite directory (you will need to make a new folder – e.g. Power) and then create a new folder under that (perhaps ‘power’) and then add a ‘package’ statement to the program (e.g. package power;)

Use evosuite to **generate tests for power.java**.

### Handin

\*\* (a) Take a screenshot of the 'statistics.csv' file

\*\* (b) Provide a screenshot(s) of the test file

\*\* (c) Has the faulty branch from power(int,int) been covered? If so, does it find the fault? If not, explain

```
if(right <=0)
```

```
{  rslt=-1; }
```

6. Now **manually copy the test inputs from evosuite** into a Junit format that can be used by the jacoco tool. To do this, create a new PowerTest.java class inside of the jaccoco test directory and populate the tests using the inputs and oracles from the evosuite tests. Run the jacoco tool and get a code coverage report.

### Handin

\*\* compare the code coverage report with the one you got for assignment one (the coverage before you fixed the faults and after you added new coverage). **Include a screen shot** of the coverage report for the evosuite tests and discuss the difference from your original, manually created coverage (if any).