**HW 01: Testing triangle classification**

SSW-567 – SW Testing, Qual. Assur. & Maint

Srikanth Uppada

*“I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.”*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Deliverable 3**:

* What challenges did you encounter with this assignment, if any?
* What did you think about the requirements specification for this assignment?
* What challenges did you encounter with the tools?
* Describe the criteria you used to determine that you had sufficient test cases, i.e. how did you know you were done?

1. What challenges did you encounter with this assignment, if any?

**Ans:** Initially the approach was to test the requirement. Based on the requirement few test cases were written to identify any bugs in the function classify\_triangle(). Once the test cases results are generated, it is evident that the requirements specification is incomplete.

So during test case development, it was not clear what to expect for some corner case scenarios. Also the usage of pytest framework is a new learning experience for me. However the assignment helped to understand the concept of unit test on a particular program.

1. What did you think about the requirements specification for this assignment?

**Ans:** The requirement specification was incomplete, It lacks complete information to classify a triangle based on lengths of sides. Also some corner cases scenarios are required to have a complete requirement specification. Like what should the program return, if the inputs are invalid.

1. What challenges did you encounter with the tools?

**Ans:** Using the pytest framework is new for me, It is a learning experience. In the beginning the PyCharm run configuration was defaulted to unit test framework instead of pytest. So had to spend some time on it to identify why the test cases were not executing. Also missed the prefix “test\_” for the test cases and test case class. Overall some effort was spent to execute the test cases successfully.

1. Describe the criteria you used to determine that you had sufficient test cases, i.e. how did you know you were done?

**Ans:** The criteria was to have a nominal and corner cases to verify the program classify\_triangle(). But after the test results, few gaps in test cases and requirement were identified, which lead to the modification of requirement specification and the program.

Also for each and every test case the inputs are maintained to have different scenarios instead of repeating the same scenario. By following this approach, all the possible test scenarios are covered. Anything more than this would be repetitive test case. Also the test cases were only written to cover all possible scenarios based on the data type. Any change in data type might require additional test cases. In our case only integer type is considered