Texas Tech University
Department of Computer Science
CS5374 – Software Verification and Validation
Spring 2025, Individual Assignment 1
Due Date: Feb 28 (Midnight)
Submission through the canvas system only
9 marks for on-campus, 10 marks for distance
Note. The total mark in the assignment is 10 that will be adjusted for on-campus students.

Goal 1: 1) Designing and Implementing Functional/Structural Tests; 2) Practicing tracing a Langgraph project with LangSmith (8 marks)

Part A:
Designing and Implementing Functional/Structural Tests

Consider the following Github repository:

https://github.com/AutomationPanda/shopping-cart-unit-tests

The repository contains source code for implementing a shopping cart (i.e., order) written in Python. The repository also contains some test for the project.

As a tester, you need to familiarize yourself with the code.

Objective:

1. Install the code, and execute the tests provided by the repo in the following folder
2. Design and implement 20 functional tests based on Equivalence Partitioning
3. Design and implement additional structural testing using Python or Studio code coverage tool
4. Measure the adequacy of testing for the given repo

Deliverable (provide snapshots for each stage):

1. Install the code and provide some snapshots on execution of the code. (1 mark)
2. Execute the tests provided by the repo and provide the snapshot of execution. (1 mark)
3. Measure the adequacy tests of the tests based on statement coverage using tools such as Coverage.py (1 mark)
4. Design the specification of 20 functional tests using Equivalence Partitioning in the tabular format given below. (2 mark)
5. Implement and execute the 20 functional tests designed using EP using unittest unit testing framework or similar unit testing tools. (1 mark)
6. Measure again the updated adequacy tests of the newly added functional tests based on statement coverage using tools such as Coverage.py (0.5 marks)
7. Design and implement structural tests for increasing statement coverage to 80% for the entire project (1 marks)

8. Measure again the updated adequacy tests of the newly added structural tests based on statement coverage using tools such as Coverage.py. The statement coverage should reach 80/100% of reachable code. (0.5 marks)

-

| Source code .py | TC# | EP class | Valid/Invalid ? | Test inputs | Expected output | Status pass/fail |
|---|---|---|---|---|---|---|
| … | | … | … | … | … | … |
| … | | … | … | … | … | … |

Part B: Practicing tracing a Langgraph project with LangSmith

Implement quiz 1 in Langchain, and trace the execution using LangSmith.

Deliverable:

1. Source code of the Langsmith version of quiz 1 with tracing enabled. (0.5 mark)
2. Execution of quiz 1 on screen and a screenshot of the output (0.5 mark)
3. Report and provide screenshot(s) of tracing and the number of chains executed during one time execution obtained using LangSmith (1 mark)