SOFTWARE VERIFICATION, VALIDATION AND TESTING

ASSIGNMENT 1 - Software Unit Testing Framework

Purpose

This project aims to showcase your understanding of unit testing by utilizing appropriate tools, creating effective unit test cases, and critically evaluating the use of an AI-powered language model for unit-level testing.

Objectives

By completing this assignment, you will:

- Apply unit testing principles to design and execute tests for an algorithm.
- Utilize a generative AI tool to develop unit test cases within a unit testing framework.
- Execute and analyze the performance of the test cases and refine them for improved coverage and accuracy.
- Evaluate the effectiveness of a generative AI tool in unit-level testing.

Project Description

This assignment focuses on using a unit testing framework to develop and execute test cases for an algorithm, incorporating generative AI tools to enhance the process.

Tasks

1. **Algorithm Implementation**
    - Develop or download code for the Heapsort algorithm in a programming language of your choice (e.g., Java, C++, Python, or JavaScript).

2. **Unit Testing Framework and AI-Assisted Test Generation**
    - Research and select an appropriate unit testing framework to validate your Heapsort implementation.
    - Use a generative AI tool to generate unit-level test cases within the chosen framework.
    - Document the prompts used to interact with the AI tool and the resulting test cases it generates.

3. **Test Execution**
    - Execute the generated test cases in an Integrated Development Environment (IDE) of your choice.
    - Provide evidence of test execution and results, such as screenshots of the output.

4. **Test Case Improvement**
    - Evaluate the adequacy and validity of the AI-generated test cases.

- Refine and enhance the test cases based on their performance and coverage to ensure they comprehensively test the algorithm.

5. **Assessment of Generative AI Tool**
    - Reflect on your experience using the generative AI tool for this assignment.
    - Analyze the tool's effectiveness in generating unit-level test cases, highlighting its strengths and limitations based on your observations.