Ira A. Fulton Schools of Engineering
Arizona State University

CSE 565: Software Verification and Validation

Structural-Based Testing Assignment

Purpose

The goal of this project is to explore and analyze code coverage using structural-based testing techniques.

Objectives

Learners will be able to:

- Analyze code coverage using statement and decision coverage techniques.
- Develop a set of test cases based on specified requirements.
- Learn about different data flow anomalies.

Technology Requirements

- Java 17 or above

Project Description

In this assignment, you will focus on two vital aspects of software testing and code analysis. The project consists of two distinct parts, each geared towards honing your skills in specific areas.

In Part 1, you will explore code coverage analysis for a Vending Machine application. You will learn how to assess code coverage using statement and decision coverage techniques, enabling you to identify untested areas and enhance the test suite's efficiency.

In Part 2, the focus shifts to data flow analysis in Static Analysis code. You will develop the ability to detect data flow anomalies, such as data leaks and uninitialized variables.

Directions

Download the provided files located in the assignment page, then carefully review the directions before starting your work:

- VendingMachine.java
- StatisticAnalysis.java

Part 1:

You are expected to **research and experiment with a tool** that provides statement and decision code coverage.

You are given a Java file called **VendingMachine.java** and you are asked to **develop a set of test cases** for this code based on the following requirements:

- Takes in an integer input
- Allows users to select between three products: Candy (20 cents), Coke (25 cents), Coffee (45 cents)
- Returns the selected product and any remaining change
- If there is not enough money to buy the product, displays the amount necessary to buy the product and other products to purchase.

You are expected to **execute** the program with your test cases, **observe and report out the code coverage** of your test cases. The goal is to reach **100% in statement and 90% in decision coverage**. You should be changing your test cases until you reach the desired coverage

Part 2:

You are expected to **research and experiment** with a static source code analysis tool.

You are given a Java file called **StaticAnalysis.java** which contains two different data flow anomalies.

The inputs to this code are:
- the weight of the package as an integer
- the length of the package as an integer
- the type of product as a String

You are expected to **execute the tool** you selected and **analyze** the report generated for StaticAnalysis.java code. You need to analyze the findings of the tool on this code and comment on how well the tool has performed in identifying the two built in anomalies.

FInally, you are expected to assess the tool in your own words in terms of:
- Features and functionalities provided by the tool

- Type of anomalies covered by the tool
- Ease of use

Submission Directions for Project Deliverables

You will submit all project deliverables compiled into **one (1) report in PDF format**.

Title your file as **yourlastname_firstname_CSE 565_StructuralBasedTesting** and submit at the assignment submission page.

A single report in APA/MLA style should include Part 1 and Part 2 consisting of all the screenshots of test cases, coverage and detailed explanations about what the screenshots represent.

Part 1

It should include:

1. A description of the tool used and the type of code coverage it provides
2. Description of the set of test cases developed including the screenshots of the code written
3. Report and discuss the coverage achieved for the test cases executed by including screenshots showing the tool's coverage

Part 2

It should include:

1. Description of the tool used and the types of analysis it provides
2. Description and analysis of the anomalies detected by the tool, showing screenshot of the report generated
3. Your assessment of the tool in your own words in terms of:
    - Features and functionalities provided by the tool
    - Type of anomalies covered by the tool
    - Ease of use

Submission Guidelines

You may submit your deliverables as many times as needed. However, only the most recent submission will be graded.

You must submit your Project 2 file in the designated submission space. Learners may **not** email or use other means to submit any project for review, including feedback, and grading.

Evaluation

You will be evaluated on the criteria for each part (**worth 100 points total**):

Part 1

1. **Description of the tool and type of coverage (10 points)** - Description of the tool used and the types of coverage it provides
2. **Development and description of a set of test cases (20 points)** - Test cases covering all code coverage requirements. You are expected to include a screenshot showing the test cases developed for achieving the required coverage.
3. **Reporting out and discussion of the code coverage (20 points)** - You are expected to include a report and discuss coverage achieved by the executed test cases by including screenshots showing code coverage achieved by the test cases

Part 2

1. **Description of the tool and type of anomalies covered (10 points) -** Description of the tool used and the types of anomalies it covers
2. **Description and analysis of the anomalies detected by the tool (20 points)** - Description and analysis of the anomalies detected by the tool, showing screenshot of the report generated
3. **Assessment of the tool (20 points)** - Your are expected to evaluate the tool in terms of:
    - Feature and functionalities
    - Coverage provided
    - Ease-of-use