

**Due:** Friday 16 May, 23:59 GMT+8  
**Weight:** 30% of the unit mark for CMPE3008  
(35% for CMPE5000)

## Introduction

In this assignment, you will be investigating real-world code and its associated tests in a team of maximum 4 people. The goal is to apply concepts taught in Software Engineering Testing to Analyse the code, evaluate the testing suit in place, and propose improvements.

## The Tasks

As part of this assignment, you are expected to complete a series of tasks. You are first tasks with finding source code according to specifications detailed below. Using that source code, you are to then investigate and apply coverage techniques.

### Source Code

Part of this assignment will be to find suitable source code for use with this assignment. There are many public repositories such as [GitHub](#) and [Google Open Source](#) that may assist you in finding examples. For the purposes of the assignment, you are able to choose code from any exception-capable language you wish (so long as it is not an esoteric language such as [Whitespace](#).)

The source code *must not* be a previous Curtin assessment that you or another student have written.

### Requirements for Source Code

You are to find source code with at least **two (2) methods** and at least **two (2) related test methods** for a total of four methods. Additionally, your choices should adhere to the following requirements.

- All chosen methods:
  - Must not be empty
  - Must come from a publically available source
- All Non-Test methods:
  - Must have some form of control structure, such as if statements and/or loops
  - Must contain at least five lines, not to include method declarations or lines containing only brackets
  - Must have at least one variable
  - Must have at least one input (explicit or implicit)
  - Must have one of the following:
    - Multiple returns
    - An exception is thrown
    - A loop is present

### Coverage & Testing

Once you have chosen the methods to investigate, you are to perform the following tasks using concepts taught in the lectures and tutorials:

- Convert your non-test methods into graphs
- Perform Prime Path Coverage on your graphs
- Perform Syntax-based Testing as follows:
  - Create a set of mutants for each non-test method
    - The set must contain at least four (4) mutants per non-test method; eight (8) total
    - These mutants **must** use at least four (4) different mutation operators
  - For each mutant above, analyse its Reachability, Infection, and Propagation conditions

Compare your results to the test methods you have chosen and make a short analysis of those test methods. Note that some projects may require some complex knowledge of their context. For this reason, you are **not** expected to determine how thorough or complete the coverage of the tests are. You should, however, make an estimated judgement regarding how many test paths are performed and provide an opinion on whether or not the testing is sufficient.

As part of this analysis, you must include whether or not the tests would discover any of your mutants.

## Testing Tool Investigation & Presentation

Your team will research the tools and frameworks that are available for testing software projects, write a summary, and give a presentation providing an evaluation on a particular tool. The tool or framework you choose and report on will be at your team's discretion, but it should be related to testing software.

**You may not use JUnit or Python's unittest library for this assignment.**

Your team should identify, at a minimum:

- The purpose and functionality of the given tool or framework
- How the tool or framework should be used
  - General descriptions and examples should both be provided
- Any other relevant considerations (such as standards, popularity, etc.)
- Whether or not your team would recommend it for actual use

Although you may choose any relevant tool, a useful place to start can be found here:  
<http://softwareqatest.com/qatweb1.html>

## Presentation & Report

Your team will create a **5-minute presentation** on your chosen testing tool/framework. This presentation will be held during the normal lecture day and times on weeks 12 and 13. It is recommended investigating "lightning talks" for ideas on how to condense information down into the short timeframe.

A **short** written report detailing your findings should be included. The report should not exceed two pages except to include graphics. The report should cover the points outlined above.

## CMPE5000 Tasks

This is a task *only* for students undertaking Advanced Software Engineering Testing. Students studying CMPE3008 do not need to complete this task; no bonus marks are available for undergraduate students.

Students taking Advanced Software Engineering Testing are additionally expected to complete a short, **individual report** on the chosen source code to be submitted separately from the group report. As part of this additional investigation, students are expected to:

- Successfully compile and run the project and execute the test methods
- Include evidence that the tests are failing or succeeding
- Modify the source code and provide evidence that:
  - At least one of the tests fail
  - At least one of the tests discovers a mutant that you insert into the code

The report should describe the steps you had to undertake to achieve these results.

## CMPE3008 AI-Based Testing Tool Task

This is a *required task* for students undertaking Software Engineering Testing (CMPE3008). This task is designed to enhance your understanding of modern AI-based software testing tools.

As part of this assignment, CMPE3008 students are expected to complete a short, **team report** exploring and critically analyzing an AI-based or Machine Learning-driven software testing tool. This report should be submitted alongside your group report and presentation.

The report (maximum three pages) must address the following:

- **Overview of the AI-based Testing Tool:**
  - Choose an AI-based Tool name and type of testing it supports (e.g., unit, regression, UI, performance).
- **How AI/ML Enhances Software Testing in this tool:**
  - Describe specific AI capabilities of the tool (e.g., automatic test case generation, defect prediction, test prioritisation, visual testing).
- **Comparison with Traditional Tools:**
  - Explain how this AI-based tool improves upon or differs from conventional tools like Selenium, PyTest, etc.
- **Critical Evaluation:**
  - Discuss the benefits, risks, and limitations of using this AI-based testing tool.
  - Would your team recommend its use in real-world projects? Why or why not?

The report should present a well-structured analysis and demonstrate your team's understanding of AI integration in modern software testing tools.

## Deliverables

The assignment should be submitted as a .pdf document or series of documents. They should:

1. Include the details of your chosen code project and where it can be located
2. Your chosen methods and their associated graphs
3. The coverage for those graphs

4. Your Syntax Testing analysis
5. Your analysis for each test
6. Your written summary for the research tool

## Marking

The allocation of marks for this assessment are as follows:

- Source Code
  - Code is accessible via public repository
  - Code conforms to criteria specified
    - No specific marks, however, penalties apply to graph-related coverage for missing criteria
- Coverage & Testing
  - Prime Path Coverage
  - Syntax-Based Testing
  - Test Analysis
- Test Tool/Framework
  - Written Report
  - Presentation
- [ CMPE5000 Report ]
- [ CMPE3008 Report ]
- Submission Quality
  - Submitted materials are professionally laid out, readable

## Academic Integrity

Please see the *Coding and Academic Integrity Guidelines* on Blackboard.

In summary, this is an assessable task. If you use someone else’s work or assistance to help complete part of the assignment, where it’s intended that you complete it yourself, you will have compromised the assessment. You will not receive marks for any parts of your submission that are not your own original work. Further, if you do not *reference* any external sources that you use, you are committing plagiarism and/or collusion, and penalties for academic misconduct may apply.

Curtin also provides general advice on academic integrity at [academicintegrity.curtin.edu.au](https://academicintegrity.curtin.edu.au).

The unit coordinator may require you to provide an oral justification of, or to answer questions about, any piece of written work submitted in this unit. Your response(s) may be referred to as evidence in an academic misconduct inquiry.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21