CPS3230 Assignment: Part 1 – Unit Testing

Deadline: Monday 6th January 2020 at noon

Instructions

1. This assignment will be marked out of 35 and accounts for 35% of the study unit mark. It is estimated that it will take between 20 and 25 hours to complete this assignment.
2. This is the first part of a three-part assignment.
3. The work is to be done on an individual basis, not in groups.
4. The assignment is to be submitted via VLE by noon of Monday 6th January 2020.
5. A random sample of students will be interviewed about their assignment.
6. Your final submission should consist of a ZIP file containing (1) a PDF file documenting your efforts and decisions; and (2) any code you developed as part of the assignment.
7. In your documentation, be sure to include a link to the git repository containing your code.
8. Commit code to your repository regularly (with helpful commit messages) so that the examiner can understand how your work evolved throughout the task.
9. Your final submission should include work related to all three parts of the assignment.

Overview

You have joined a new company and inherited a project that has no documentation or unit tests. The system, written in Java, deals with foreign exchange and after running the main class you notice that the menu allows the user to:

1. List currencies
2. List exchange rates between major currencies
3. Check exchange rate
4. Add currency
5. Delete currency

Your manager explains that the system is not overly complex but does make calls to a server to obtain exchange rates, relies on timing for optimisation, and makes use of files for saving currencies to disk. You are being tasked with developing a unit test suite around it to help you and future developers.

Requirements

You are required to carry out THREE tasks as outlined below.

Task 1 – Setup (5 marks)

1. From the VLE, download ***currencymanager.zip***, which contains the source code associated with this assignment.
2. Create an online git repository and upload the source code to it.
3. Invite mark.micallef@um.edu.mt to be viewer on the repository.
4. Understand the system by using it as an end-user (run CurrencyManager.java).

Task 2 – Initial unit tests (10 marks)

1. Develop unit tests for the system-under-test (SUT) without making any changes to the system itself.
2. Document the statement coverage you achieve and concretely explain the reasons why certain code areas are left uncovered by tests.
3. Comment on any testability issues present in the SUT.

Task 3 – Advanced unit testing (20 marks)

1. Gradually refactor your system to make it more testable and add more tests to cover uncovered code sections as you do this. Push to git regularly using useful comments in the commit to explain the change you are pushing.
2. Use and document at least one instance of *Dependency Injection* pattern and at least on instance of *Dependency Lookup* pattern.
3. Report your new coverage metrics and explain how improvements were made and what/why statements, if any remain uncovered.