Testing, testing, testing with TDD

Objectives

- Writing tests that fail (RED)
- Writing code that passes those tests (GREEN)
- Writing cleaner code that does not break those tests (REFACTOR)
- Continue practice of Git/GitHub

USING SEG3103_PLAYGROUND

- Create **/lab04** directory
  - Extract **tic.zip** and **fizzbuzz.zip**
  - Make sure you can
    - compile code
    - Run tests

Now ignore those projects :-)

- Create a new project in Java, Elixir, or Python
    - Pick a project to "TDD" on. If you pick tic-tac-toe or fizzbuzz you CANNOT use Elixir
- Practice test-driven development
    - Write a small desire (failing test)
    - Have it fail (commit that change)
    - Have it pass (commit that change)
    - Make it better if relevant (and commit that change)

Test Drive For An Hour

- Once you have a your first failing test start the clock
- Red - Green - Refactor for an hour
  - If you encounter a technical challenge stop the clock
  - If you get stumped with a Java (or JUnit issue) then stop the clock
- Every progress forward should be captured as a commit (this includes adding a new RED failing test)

Submission

- All work should written under
  - **seg3103_playground/lab04**
- Git commit message should include the results of the JUnit tests
- Create **README.md** to summarize your work
  - Identify at least 5 commit groups (per person) to highlight examples of
    - A failing test
    - A passing test
    - Refactored code (where the test still pass)
  - To clarify, if working in a team then you need10 commit groups in total
    - 5 per teammate
    - Clearly identify otherwise you will be penalized
- Share your repository with the teacher and TA (s)
  - Submissions to BrightSpace should clearly reference your GitHub repository