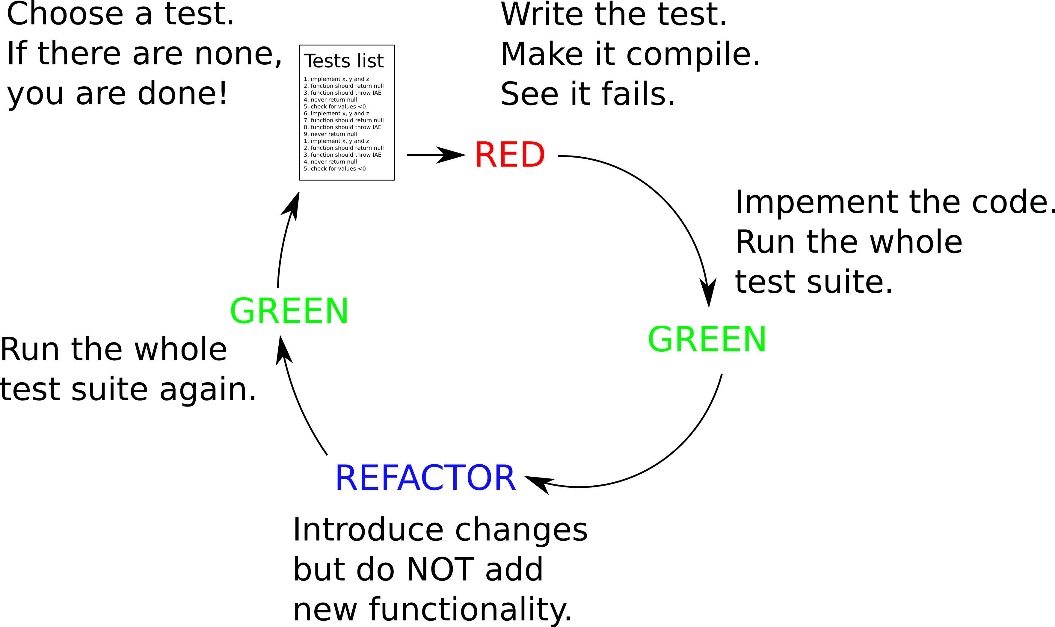
# TCI – week 3 practical

These exercises should be executed individually. All these skills are needed in the final project.

## TDD

### 3.1 Assignment – Using TDD to create justifiably correct code.

**Description**



Make sure you understand TDD! See chapter 4 of the book, sources in sheets and below.

Given is the description below of the requirements of a piece of software. It’s your job to use TDD to create the code. You can determine the order in which you will implement the requirements, explain your decision.

COMMIT TO GIT AFTER EACH TDD CYCLE!   
The history of your commits is used as proof you used TDD as a development approach. No history, no result (this is also how you need to work in the final project)

A requirements description is provided in document

'PA3\_3.1\_RequirementsToCreateTestsFor.docx'

Translate the requirements into needed unit tests using TestCherry. Ask each other and the teacher if requirements are unclear. Keep track of assumptions when you decide on what tests you need.

Use TDD to create tests first & implement each test one by one. Commit to git after each creation of a test, and after each implementation belonging to that test.

**Constraints**

Use a TDD approach.

Use TestCherry for translating the requirements into tests.

Run a code coverage reporting after each TDD cycle. Save all coverage reports because you will have to show them.

Note: this assignment probably will overflow into next week. That's intended.

**Sources for inspiration**

Extensive and really insightful example of using TDD, unit testing, parameterized tests etc. on Game of Life: <https://www.youtube.com/watch?v=6rp43uZhuzE> (guy has a bit of monotone voice, so definitely be awake when you start)

using TDD in IntelliJ: <https://www.youtube.com/watch?v=QDFI19lj4OM>

or in written form: <https://www.jetbrains.com/help/idea/tutorial-test-driven-development.html>

======================= End of practical. =============================