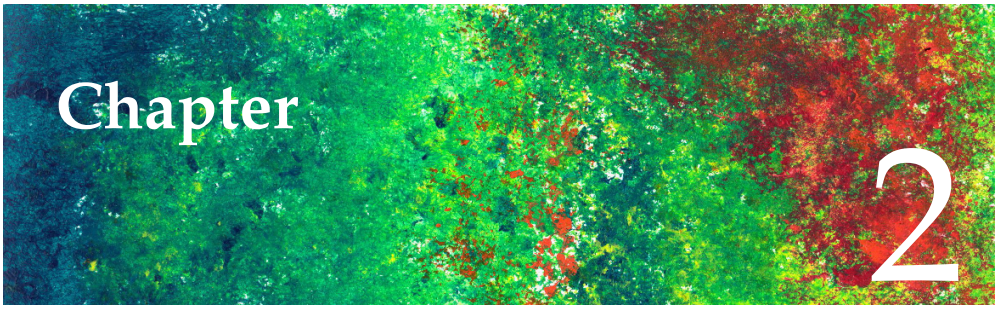

Advice for Each Examination Area

2.1 Regarding Test-driven development

2.1.1 Regarding the theory

The central TDD theory consists of a) the values b) the rhythm and c) the TDD principles. And these again rely on the testing terminology of FRS Chapter 2.

It makes a good impression to be able to explain the rhythm and some of the central principles and especially if you can relate them to the values of TDD.

2.1.2 Regarding solving the exercise

Be sure to read the exercise so you solve the right exercise and not what you think is the exercise.

Do not make an overly elaborate test list—focus instead of making the first two or three iterations. This will usually get around central TDD principles like One-Step-Test, Fake-it and Triangulation, and of course the rhythm.

You *will* be required to write proper JUnit code, so be sure to train the syntax of @Test, assertThat(), @BeforeEach, etc.

One issue I see quite often is a presentation of iteration 1 (which of course is fake-it-till-you-make-it code) which then leads to a iteration 2 *that does not contribute any production code!* Other pitfalls are iterations that *produce production code not driven by the tests* or *loosing focus, writing tests for different parts of the specification.* Consult the Kata on one of the last weekplans for examples.

Needless to say, this is a "failure in understanding the core of TDD", which influence the grade negatively.