Individual Assignment COMP4004
Due: October 21st 2022 at 7AM in Brightspace

Piraten Kapen (or simply Pirates!) is a game whose rules are posted. You are to implement a **networked** (client/server) version of this game, with a **text-based** interface that allows each player to see the current score of ALL players. Your code must support playing a single **non-rigged** (i.e., random rolls and random fortune card) game of **3000** points (to make it shorter) for exactly 3 players. The first player will initiate the game, then the others will join.

You are to use IntelliJ, Java, JUnit, Maven and a ***private*** Github repository (to which you will invite jeanpier-prof and **all** TAs) to develop your solution using a TDD approach. **The name of your repo must be 4004-F22-A1-<yourStudent#>.** TDD focuses on *unit* testing. It requires that you write and commit (in a U-TEST commit) one or a few tests pertaining to a single procedure and **then** commit the code to make these tests pass (in a separate CODE commit). You are to also address *acceptance* testing of this game using the test suite whose first version I will post Sept 28. (A second version *may* be posted later.) This test suite must be implemented using JUnit. You are to use a A-TEST commit for each of my test cases. Unit-level testing and coding may precede OR follow an A-TEST commit. **Most importantly**, we must be able to run my acceptance test suite in one shot (as opposed to test case by test case). Note that you may use REFACTOR commits.

Acceptance tests will require that you '**rig'** your game. That is, in order to run each of my acceptance tests, you will have to 'fake play' the actual game by replacing random outcomes by the outcomes specified in a test. In particular, most of my test cases are single-player ones that will call the code to draw a fortune card and roll and rerolls dice BUT THEN set that fortune card and dice to specific values (given in that specific test case), compute a score and assert it is the same as the expected one. For a multiplayer test, you must log player behavior to a file we will manually check.

If we do not have access to your repository, or if it is public, or if you do not use the specified tools and a TDD approach, or if you do not follow rules for commits, automated testing, and submissions, you will get a mark of zero.

Submission
You will submit a single file, which consists of a filled **correction grid** to Brightspace.
Follow the instructions posted with it.
This correction grid defines a multitude of tests ***mainly*** pertaining to scoring the roll or successive rolls of a single player whose turn it is.