Homework 1

You have to submit your solutions as announced in the lecture.
Unless mentioned otherwise, all problems are due 2017-02-23, before the lecture.
There will be no deadline extensions unless mentioned otherwise in the lecture.

Problem 1.1 *Goto fail*   Points: 10

Consider the 'goto fail' example from the lecture notes.

1. Transcribe the faulty function (including the fault) as closely as possible in some programming language. Because we will use unit testing and code analysis tools, it is good to use a language for which good tool support exists, e.g., C(++) or Java.

   Replace all types, functions, and variables that are declared outside the relevant code snippet with arbitrary implementations. The only thing you have to ensure is that all three function calls in the body can succeed or fail independently—i.e., it should be possible to achieve all eight combinations of which of these function calls succeed/fail by choosing appropriate inputs.

2. Write a unit test for your function.

   The test cases should be such that (if the function were implemented correctly) running your tests executes every line of the tested unit at least once.

   Thus, you need at least 4 test cases to cover all 4 possible paths the function can take. However, it is better to have 8 test cases in order to cover all 8 combinations of which function calls succeed/fail.

3. Put your program on github and set up a travis test to automatically run your tests on every push.

4. Setup a code coverage library/tool to use with your program.

   Run your unit test and see if every line is executed. That should detect the fault (because the faulty implementation will not execute every line).

5. Setup a static code analysis tool to use with your program.

   Run an unreachable-code-analysis. That should detect the fault.

6. Setup a code style checker (e.g., as part of your IDE) to use with your program.

   Configure it such that irregular indentation is flagged. That should detect the fault.

Regarding the last 3 questions:
- Depending on your chosen programming language, you may have to try multiple libraries/tools until you find one that works well.
- You are allowed to help each other by circulating tips which libraries/tools work well for which programming language. But everybody has to set them up and run them herself.
- When submitting your homework, submit the log output, a screenshot, or something similar that shows how the fault is detected.