Exercise Sheet: Software Testing

1. Unit Tests for a Class Stack

Create JUnit test cases for the provided Stack class, which manages non-negative integers in a stack. Proceed as follows:

a) Create appropriate test cases for the methods isEmpty(), push(int), pop(), and size() in a test class called StackTest. In particular, test the exception handling of the push(int) method.

b) Add a method public int top() to the Stack class that returns the top element of the stack or -1 if the stack is empty. To so in a test-driven manner and first create tests for that method and then implement the top method itself.

Upload the adapted class Stack as well as the class StackTest.

2. Control Flow Coverage

Provide control-flow-based tests for the method gcd oft he provided class MyMaths to compute the greatest common divisor of two positive integer numbers. Perform the following steps:

a) Create a control flow graph for the method gcd.

b) Design a minimal test case set (as done on the lecture slides) for statement coverage.

c) Design a minimal test case set (as done on the lecture slides) for branch coverage.

d) What is the issue with path coverage for gcd? Figure out what boundary interior coverage is, define it and design test cases for the method gcd that fulfil boundary interior coverage.

e) Implement the test tests that you created in task b) und c) in IntelliJ and check via „Run with Coverage" whether they fulfill statement coverage and branch coverage.

3. Condition Coverage

Create test suites to cover the condition (!A || !B) && (A || B) with

a) Simple Condition Coverage,
b) Modified Condition Decision Coverage,
c) Multiple Condition Coverage.