

Assignment 7

For this exercise you will need to measure statement and branch coverage. To automate your coverage measurements you can install the *Code-Cover* plugin for Eclipse from codecover.org.

Exercise 1: Statement and branch coverage

You are given the buggy implementations of two methods; see `Array.java`. The method `indexOf` returns the first occurrence of an integer in an array of integers. The method `average` computes and returns the average computed over an array of integers.

1. Write a test case that reveals the bug in the method `indexOf`.
2. Write a test suite that achieves 100% statement coverage for `indexOf` and does not find the bug.
3. Is there a test suite that achieves 100% branch coverage for `indexOf` and still misses the bug?
4. Write a test case that reveals the bug in the method `average`.
5. Write tests that achieve 100% branch coverage for `average` and miss the bug.

Exercise 2: Functional vs. structural testing

Knapsack Problem: Given a set of items, each with a mass and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

An algorithm's implementation that solves the Knapsack problem is given in `Knapsack.java`.

1. You are given an initial test suite. Measure the branch coverage achieved by the test suite. Add tests until you get 100% branch coverage, and fix any bugs that you find.

2. Derive a test suite that exercises the functional behavior of the knapsack problem's implementation, and fix any bugs that you find.
3. Discuss the advantages and disadvantages of structural and functional testing.