Software Testing MTAT.03.159
Lab 3: White-box Testing

Lecturer: Dr. Dietmar Pfahl
TA: Briti Deb
TA email: deb@ut.ee
Inst. of Comp. Science, University of Tartu

April 2013

- Submission deadline: Before the start of the next lab (i.e., before start of Lab 4).
- Late Policy: 50% deducted for every 24 hours late
- Group: Maximum two members in a group. Answers should be your own group work, in your own words.
- Maximum eight points.

1 Introduction

In white-box testing, the purpose is to ensure that all the code has been tested and covered. There are different coverage measures depending on the level of detail of the white-box test. The coverage measures may also be applied to integration testing and system testing.

2 Learning Objectives

The exercise aims at giving an understanding of white-box testing. The specific learning goal is to gain an insight into the concept of white-box testing using coverage measures.

3 Preparation (on paper)

Assignment 1: Read chapter 5 in [1] and the slides from the lecture White-box test techniques.
Assignment 2: Look at the pseudocode below. How many feasible paths are there? Write a set of test cases that gives you 100% coverage of all the feasible paths.

```
### Pseudocode ###
input (score);
if score < 45 then
print ('fail');
else
print ('pass');
if score > 80 then
print (' with distinction');
end
```

Assignment 3:In this lab session you will test the NextDate program with white-box techniques. Get an understanding of how the program works, the code is available at the course web-site.
Assignment 4: Make flowgraphs of the methods in the NextDate class.
Assignment 5: Prepare test cases based on the flowgraph. Discuss the use of different

coverage measures, such as Statement Coverage, Branch Coverage, Path Coverage, etc.

4 Exercise (on computer)

Assignment 6: Now, test the NextDate program using white-box testing. Implement the testcases you have prepared on paper using coverage measures such as Statement Coverage, Branch Coverage, Path Coverage, etc. Implement more test cases if you realize that you missed any during the preparation.

Remember to specify test case ID, what is tested, input, expected output and other useful information while writing your test cases. You may also want to make room for pass and fail notes and perhaps for comments. Record your test results carefully for your test report.

5 Report

The report should be submitted via the course web-page using the 'submit' button – selection: Lab 3

On the first page of your report, write the name of the lab, group member's names, and email addresses. The expected size of the report is 4-5 A4 pages, and should focus on conclusions from the lab session.

The purpose of the report is to discuss the result of the exercise and related topics. Describe the outcome of your tests. The report should include the following five sections.

- The flowgraph. (1 point)
- The test cases (see example test case in Appendix 1). (5 point)
- Defects detected. An example defect report can be found in [1, p. 363]. (0.5 point)
- Which coverage criteria works best and why? (1 point)
- Compare black-box test techniques (used in Lab 2) and white-box test techniques. Discuss advantages and disadvantages with each and when they are appropriate to use. (0.5 point)

References

[1] Burnstein, I., Practical Software Testing - A Process-Oriented Approach, Springer-Verlag, 2003.

[2] http://www.codign.com/pathbranchcode.html

Appendix 1

Test ID	Method under test	Coverage Criteria	Expected Result	Actual Result	Pass/Fail	Comments
1	date.run(1,0,2001)	D,C				