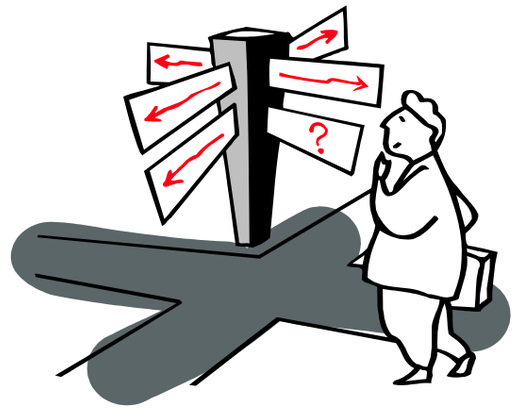
Software Testing

0. Practical Matters

0. Practical Matters



- Contents
- Goals
- Course Material
- Plan
  - Lab Sessions
  - Flipped Class Room
  - Industrial Guest Lectures
- Exams and Grade

Contents

The student will acquire experience with thorough testing and verification of a software system to guarantee with a certain degree of confidence that a given software system meets its specification.

The course has a practical ring to it with

- a minimal theoretical content
  - taught as testing best practices and involving self-training
- several lab sessions
  - trying out several test techniques and strategies on an existing representative software system
- a few guest speakers from industry
  - confirming that the testing techniques covered in the lectures indeed are used in practice

Attendance is required.

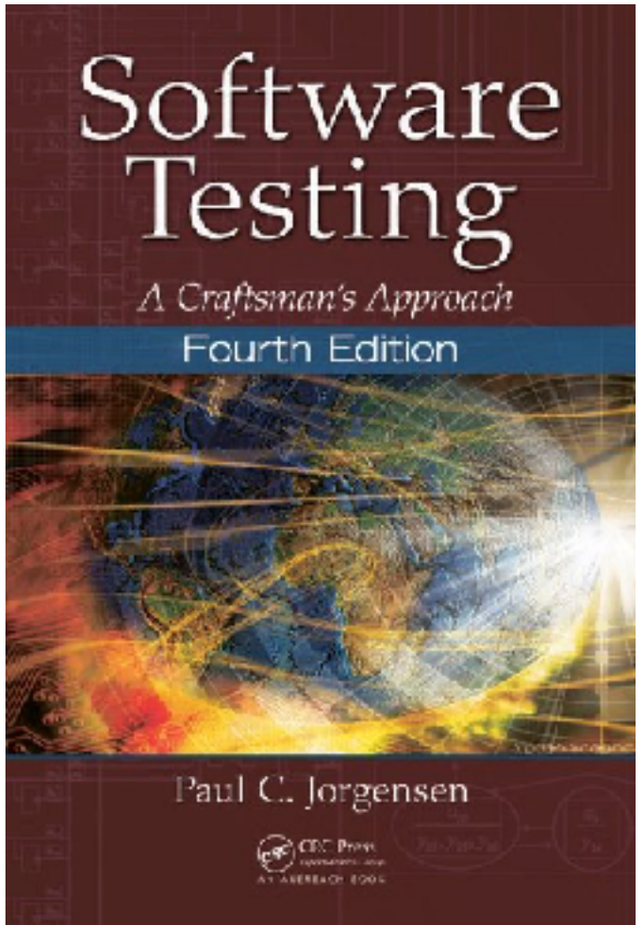
Goal(s)

You will be able to …

- apply white-box and black-box test techniques to build a test-suite;
- assess (and improve) the coverage of a test-suite;
- distinguish between various test automation strategies;
- select the most appropriate test techniques for a given test strategy.

Focus is on
Test Automation

Course Material



Buying the book for the course is not necessary.
Electronic copies are available via the library (VPN access required).

Precondition check

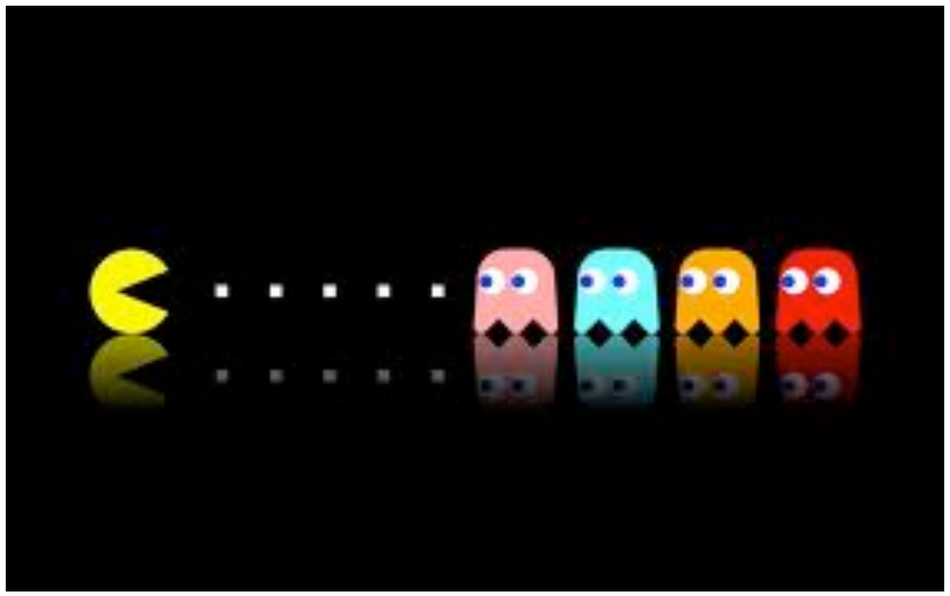
(Ideal) Schedule

Detailed planning @ course web-site

1	[T] Introduction	prof. Demeyer
2	[T] Test Design 1	prof. Demeyer
3	[T] Test Design 2	prof. Demeyer
4	[F] Test Automation: 41 Reasons	flipped class room
5	[F] Test Automation: Setting Up	flipped class room
6	[F] Test Automation: Test Architecture	flipped class room
7	[F] Exploiting Automated Tests	flipped class room
	-- easter holidays	
8	[G] Industrial guest lecture	2-3 guest speakers
9	[G] Industrial guest lecture	2-3 guest speakers
10	[G] Industrial guest lecture	2-3 guest speakers
11	[G] Industrial guest lecture	2-3 guest speakers
12	[G] Industrial guest lecture	2-3 guest speakers
13	[G] Industrial guest lecture	2-3 guest speakers

Lab Sessions

jpacman
© Prof. Arie Van Deursen from TUDelft.
https://github.com/avandeursen/jpacman-framework-jroosen



1) Introduction

2) Java testing tools

3) Category Partitioning and Boundary Values

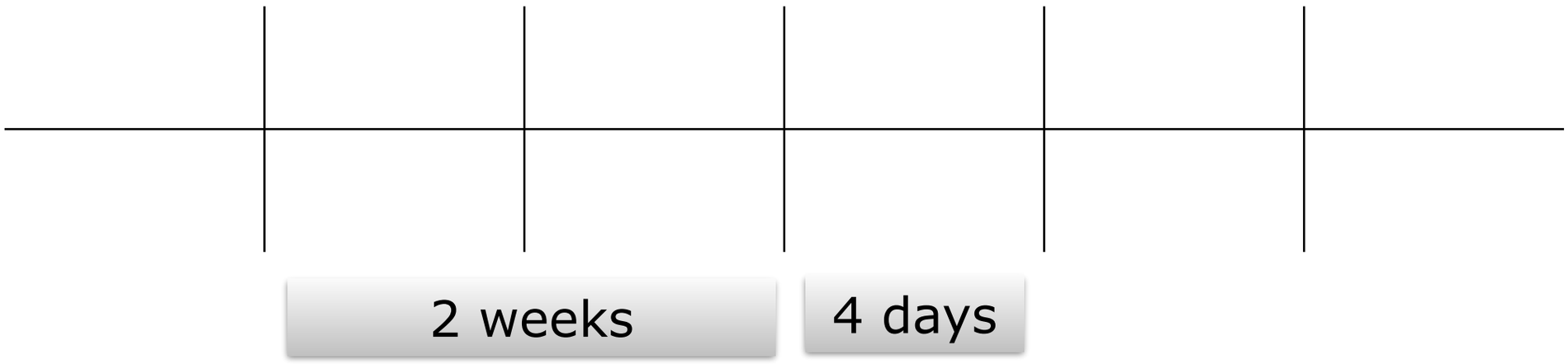
4) Decision Structures

5) State Machines

6) Mutation testing

7) Fuzzing

Lab Sessions — Timing



Release Assignment N

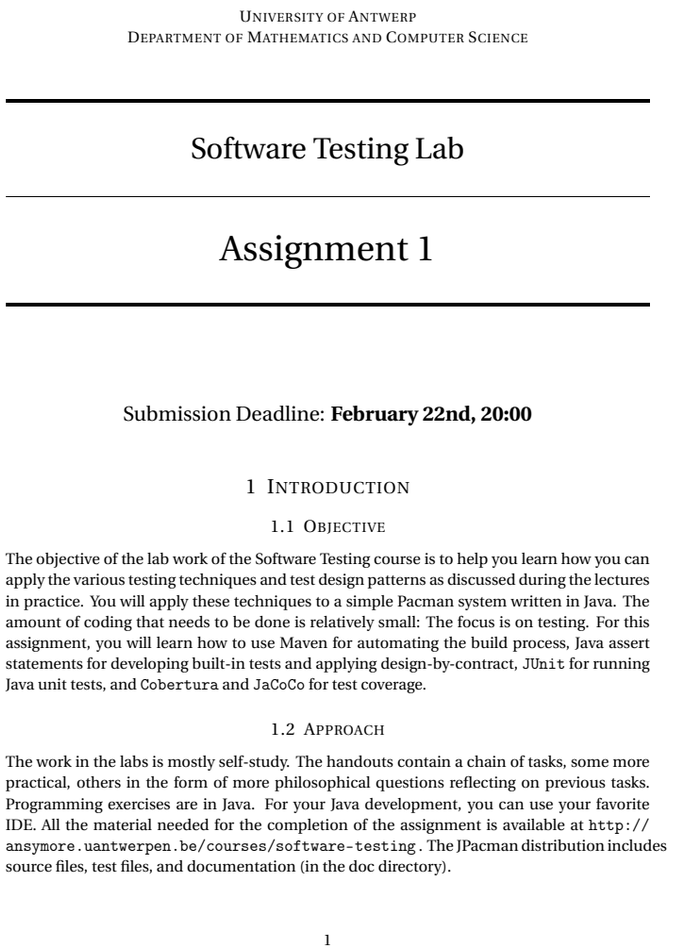
Submit Assignment N

Monday evening 20:00

Feedback Assignment N

Thursday morning 10:00

Assignments - Load



University of Antwerp
Department of Mathematics and Computer Science

Software Testing Lab

Assignment 1

Submission Deadline: **February 22nd, 20:00**

1 Introduction

1.1 Objective

The objective of the lab work of the Software Testing course is to help you learn how you can apply the various testing techniques and test design patterns as discussed during the lectures in practice. You will apply these techniques to a simple Pacman system written in Java. The amount of coding that needs to be done is relatively small: The focus is on testing. For this assignment, you will learn how to use Maven for automating the build process, Java assert statements for developing built-in tests and applying design-by-contract, `JUnit` for running Java unit tests, and `Cobertura` and `JaCoCo` for test coverage.

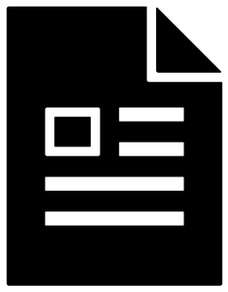
1.2 Approach

The work in the labs is mostly self-study. The handouts contain a chain of tasks, some more practical, others in the form of more philosophical questions reflecting on previous tasks. Programming exercises are in Java. For your Java development, you can use your favorite IDE. All the material needed for the completion of the assignment is available at `http://ansymore.uantwerpen.be/courses/software-testing`. The JPacman distribution includes source files, test files, and documentation (in the doc directory).

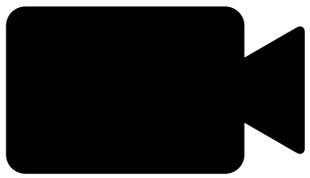
1

- 1 assignment ± 5 hours of work
- late submission = penalty
- required and optional parts
  - late submission: extra work

Flipped Class Room



1) Read background material



2) Watch video (if available)

Preparations

3) 20 minute class room discussion
4) On-line quizz

Industrial Guest Lectures

1	[T] Introduction	prof. Demeyer
2	[T] Test Design 1	prof. Demeyer
3	[T] Test Design 2	prof. Demeyer
4	[F] Test Automation: 41 Reasons	flipped class room
5	[F] Test Automation: Setting Up	
6	[F] Test Automation: Test Architecture	
7	[F] Exploiting Automated Tests	flipped class room
	-- easter holidays	
8	[G] Industrial guest lecture	2-3 guest speakers
9	[G] Industrial guest lecture	2-3 guest speakers
10	[G] Industrial guest lecture	2-3 guest speakers
11	[G] Industrial guest lecture	2-3 guest speakers
12	[G] Industrial guest lecture	2-3 guest speakers
13	[G] Industrial guest lecture	2-3 guest speakers

Attendance is required.

Exam and Grade

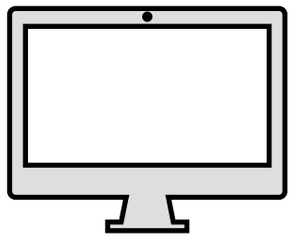
DURING THE SEMESTER (Continuous Evaluation)

- (a) Flipped class room
  + hand in quiz answers
    - punctuality
- (b) Lab sessions
  + hand in solutions via blackboard
  + at least 50/100 for each assignment

ORAL EXAM (+- 20 minutes)

- (d) Theory Lectures (random assignment)
- (c) Industrial Guest lecture (random assignment)

Use of on-line information(*)



Code

explicit attribution
of the source

Reports

Check grammar
and spelling

(*) Stack overflow, Generative AI, google, github, grammarly, …