

# Software Testing Lab

(Summer School “Software Testing and Beyond”)

---

## Assignment Guidelines

---

### 1 INTRODUCTION

#### 1.1 OBJECTIVE

The objective of the lab work of the Summer School on Software Testing course is to help you learn how you can apply the various test (design) techniques as discussed during the lectures in practice. You will apply these techniques to a simple Pacman system written in Java. Towards the end we will broaden up and also work in JavaScript. You will also be exposed to automated build systems (CI/CD), in our case maven. The amount of coding that needs to be done is relatively small: The focus is on testing.

#### 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 and JavaScript. For your Java and JavaScript development, you can use your favorite IDE.

All the material needed for the completion of the assignment is available at "Blackboard » Summer School: Software Testing and Beyond » Course Materials » Assignments". JPacman distribution includes source files, test files, and documentation (in the doc directory).

- `pacman-requirements.txt`: A text file describing the JPacman use cases
- `pacman-design.txt`: A text file describing the key JPacman design decisions

### 1.3 FRONT-HEAVY

The course and assignments are front-heavy. This means that most assignments will be given during the first half of the summer school. You are encouraged to complete and submit the assignments on time (20:00 on the day of the lab); we will in turn provide fast feedback..

**LATE SUBMISSIONS.** You are encouraged to submit the assignments on time. If you submit late, you should answer "Only when late" exercises. With these, you cannot gain points; you lose points when you do not answer them or answer them incorrectly.

### 1.4 GROUPS OF TWO INDIVIDUALS

You are expected to work in groups of two. Groups are not supposed to change throughout assignments. For each assignment, team members must share the workload (half-half), and this must be documented properly both in the code and in the report. Furthermore, each group member must examine the solution of the other member, since both group members are held accountable for all exercises.

The grouping is only for the assignments; you will be evaluated individually at the end of the summer school.

### 1.5 GRADING

Each assignment is graded from 0 to 100, and members of the same groups received the same grade. To be eligible for the final exam, you must score at least 50/100 for each assignment. Most assignments build upon each other. It is possible that some exercises can only be completed if you completed previous exercises.

### 1.6 REPORT

Please note that the report is the most important part of your answer. So take some time to write an adequate report. Your reports are expected to clearly demonstrate your solution. Describe your actions, results, and explanations for each exercise in your report in full sentences. Also provide screenshots to show that you fulfilled the exercise.

For example, Exercise 1 in Assignment 1 requires you to generate the JavaDoc and reflect on the generated report. You should include a screenshot of the main report to show that you succeeded in generating it. For Exercise 4 in Assignment 1, you may need to provide screenshots for each functional test case; you should also describe them and mention their location within the project.

Accompany the report with all of the requested material. All submitted artefacts should be stand-alone, runnable with single commands, and should not require additional packages.

Upload all files in a zipped archive with the following format.

<Surname1>-<Name1>\_<Surname2>-<Name2>\_Assignment\_<AssignmentNumber>.zip

### 1.7 THE USE OF AI-TOOLS AND ON-LINE RESOURCES

The use of various AI tools (co-pilot, ChatGPT, ...) is explicitly forbidden for the report. We will check the reports for suspicious writing style.

For the coding part, you are allowed to occasionally use on-line resources (stack-overflow, git, ...) and AI-generated code (ChatGPT, copilot, ...). The source must be explicitly attributed and you must state explicitly how you assured the quality of the code you obtained from elsewhere. You remain responsible for the code you submit!

### 1.8 QUESTIONS

There will be a lab session every afternoon to answer any questions. Prepare your questions beforehand. Broken packages, links, missing images, etc. that prevent the assignment from being completed can be reported to [kasper.engelen@uantwerpen.be](mailto:kasper.engelen@uantwerpen.be) **and** [onur.kilincceker@uantwerpen.be](mailto:onur.kilincceker@uantwerpen.be).