

Software Testing (1DL610)

HT2026 syllabus

Contacts

Lecturer

Neea Rusch neea.rusch@it.uu.se

Teaching Assistants

Behnaz Daneshkhah behnaz.daneshkhah@it.uu.se

Biruk Ambaye biruk-aklilu.ambaye.2053@student.uu.se

Gouri Hariharan gouri.hariharan.6910@student.uu.se

Course Introduction

In this course, we will be looking at software testing methodologies and their practical application in software development. In particular, we will focus on unit testing, code coverage, and white-box and black-box testing. In addition, other testing approaches and aspects will be discussed to provide breadth of understanding. The course coding assignments use the Python programming language.

Learning Outcomes

On completion of the course, the student should be able to:

- Describe key techniques and standards in software testing.
- Explain and evaluate strategies for software testing for both complete program life cycles and individual phases.
- Specify and design test cases and test programs.
- Produce appropriate documentation for test management, including test plans, test schedules and test progress monitoring.
- Develop code using test driven development.
- Be able to use suitable testing tools and discuss their range and capabilities.

Course Materials

Course textbook

Title: Introduction to Software Testing

Authors: Paul Ammann and Jeff Offutt

Year: 2017 (2nd edition)

Publisher: Cambridge University Press

ISBN: 9781107172012.

Ångström library reference:

https://uub.primo.exlibrisgroup.com/permalink/46LIBRIS_UUB/21dn0c/alma991007853859707596

Software tools

Reasonably recent versions of these tools will be sufficient for this course.

- Python programming language: <https://docs.python.org/3/>
- PyCharm IDE: <https://www.jetbrains.com/pycharm/>
- pytest: <https://docs.pytest.org/en/stable/>

Assessments and grades

The course assessments will evaluate your conceptual and practical understanding of the course topics. You must pass the conceptual and practical parts to earn a passing grade in the course.

Conceptual – Final exam

The final exam will focus on theoretical concepts. The final exam will include True/False type questions and long-form analytical questions. The grading scheme is U/3/4/5. Score of 50% is required to pass.

Practicals

There are two parts: coding assignments (2x) and a laboratory assignment (1x). Successful completion of all parts is required to pass the practical aspects of the course. All practicals are graded on a “Pass/Fail” scale. Submissions of all deliverables will be made through Studium.

Coding assignments

There will be two coding assignments. The assignments will be completed in groups of 3 to 4 students. Students are free to use their personal laptops or the university machines. You are allowed to only collaborate within your group.

Each group will be assigned a teaching assistant (TA). The group will present their work to the TA at a scheduled presentation date. It is the groups responsibility to schedule the appointment with the TA.

Each student will be graded individually based on their understanding and contribution, as demonstrated during the presentation.

Lab assignment on Test Driven Development

This is an individual lab session on Test Driven Development (TDD). You must complete this within the time period and show your results to the TAs. You will be asked to sign up for one of two TDD lab sessions (Lab 3 or Lab 4). Choose the sessions that best fits your schedule. Under exceptional circumstances, Lab 5 is a makeup lab with limited slots. You are not guaranteed to get a slot. However, it is available in case you have scheduling conflicts.

Lecture Quizzes (Ungraded)

I will post multiple choice quizzes about the lecture topics in Studium. Please use these quizzes to support and enhance your learning throughout the course.

Academic Policies

The course will follow and maintain the academic policies as defined by Uppsala University, including but not limited to the following.

Policy on cheating and plagiarism:

<https://www.uu.se/en/students/your-rights/examination/cheating-and-plagiarism>

Guidelines on the use of generative AI:

<https://www.uu.se/en/staff/organisation-and-governance/regulations/guidelines-on-the-use-of-generative-ai-in-teaching-and-assessment>

Frequently Asked Questions

Can I use Visual Studio Code instead of PyCharm?

Yes, but we will not provide troubleshooting support for VS Code.

Do I need to install anything on my personal laptop?

If you plan to use your own laptop, install the course software before the first lab.

Can I switch groups for the coding assignments?

Yes, but only swapping is allowed. Groups have been carefully formed to ensure balance. If you have serious issues, please contact the teaching assistants.

What if someone in my group doesn't contribute?

Grading is individual and based on your presentation and understanding. Please raise serious concerns with your TA.

What if I miss both labs on TDD (lab sessions 3 and 4)?

You must sign up and attend one of lab sessions 3 or 4. If you miss both, lab session 5 is a makeup slot. However, lab session 5 has limited capacity and access is therefore not guaranteed.

Are the presentations of coding assignments mandatory?

Yes, presenting to your TA is required for both assignments. You must take part in the presentation to be graded.

Can I collaborate with friends from another group?

No, collaboration is strictly limited to your own group to ensure fair grading.

Planned Schedule

All lecture rooms are at the Ångström laboratory <https://link.mazemap.com/ADtggyrH>.

Week	Date	Time	Room	Topic
45	03/11	15:15 – 17:00	101136	Lecture 1: Introduction and testing tools
45	04/11	08:15 – 10:00	6K1107	Lab: Introduction to the Course Environment (lab session 1)
46	10/11	15:15 – 17:00	10K1201	Lab: Introduction to the Course Environment (lab session 2)
46	13/11	13:15 – 15:00	4101	Lecture 2: Unit testing, test documentation, input domain modeling
47	16/11	08:15 – 10:00	101132	Lecture 3: Abstractions I, Graph coverage Assignment 1 released
47	17/11	10:15 – 12:00	101132	Lecture 4: Abstractions II, Logic expression
48	23/11	13:15 – 15:00	2001	Lecture 5: Integration and System testing
48	25/11	15:15 – 17:00	101136-A	Lecture 6: Black-box testing Due: Assignment 1
48	27/11	13:15 – 15:00	4101	Lecture 7: Test Driven Development (TDD) Assignment 2 released
49	30/11	10:15 – 12:00	10K1201	Lab: TDD (lab session 3)
49	02/12	10:15 – 12:00	10K1201	Lab: TDD (lab session 4)
50	08/12	08:15 – 10:00	4101	Lecture 8: (advanced topics) Property-based testing
50	10/12	13:15 – 15:00	101136-A	Lecture 9: (advanced topics) Testing for software security Due: Assignment 2
51	16/12	13:15 – 15:00	10K1201	Lab: TDD – makeup only (lab session 5)
2	14/1	*see Ladok	*see Ladok	Final Exam