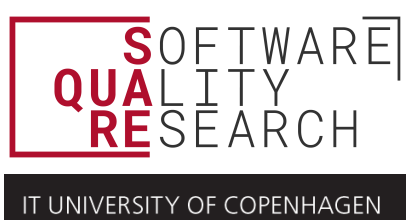
Advanced Programming, ITU 2024

After ADPRO, Theses, Exam

Andrzej Wąsowski



Advanced Programming

01. Intro (ch. 1-2)
02. Algebraic Data Types, list (ch. 3)
03. Partial Computations, option (ch. 4)
04. Laziness, lazy-list (ch. 5)
05. State monad, state (ch. 6)
06. Property-based testing (slides)
07. Property-based testing API, prop (ch. 8)
08. Parser Combinators, parsers (ch. 9)
09. Functional Design: Monad, monoid (ch. 10-11)
10. Probabilistic Programming (slides)
12. Language Semantics and Interpretations (paper)
13. Reinforcement Learning Example (mini project)

Spring'25 courses related to ADPRO

Modeling Systems and Languages (7.5 ECTS)

- Parsing, modeling, code generation, domain-specific languages, interpreters, designing own type systems
- Taught by **Eduard Kamburjan**
- We have seen parsing combinators as the internal DSL.
- Possible project: a language for robot control or digital twin model

Probabilistic Programming (7.5 ECTS)

- Bayesian stats in Python, probabilistic modeling, fit models to data
- Taught by **Andrzej + Raul**
- We have seen the basic idea in ADPRO (but we have not used data)

Spring'25 courses related to ADPRO

Advanced Programming with Types (7.5 ECTS)

- Idris, dependent types, type theory, session types (behavioral types)
- Taught by **Patrick Bahr, Marco Carbone, and Rasmus Møgelberg**
- (Way) More type glory than in Scala

Program Verification (7.5 ECTS)

- Deductive program verification with Coq (also a type theory)
- Taught by **Jesper Bengtsson**

Fall'25 courses related to ADPRO

Practical Software Analysis (7.5 ECTS)

- Automatic ways to test, fuzz, and analyze programs
- AI for fixing programs
- Taught by **Mahsa Varshosaz**
- Related to property-based testing, interpreters, evaluators, automated reasoning

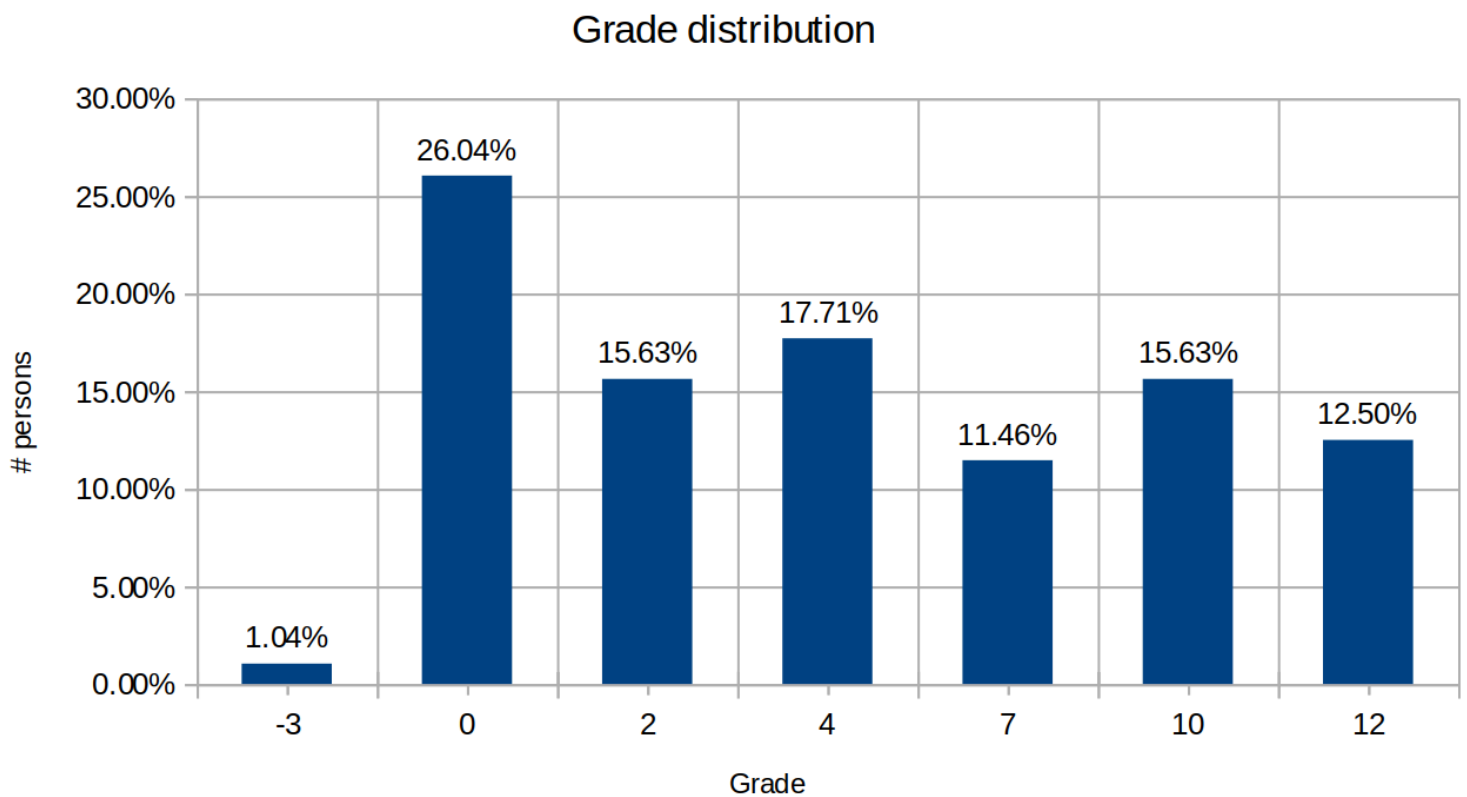
From what I know, all the above courses have project-based exams.

Thesis Projects

1. **In Software Quality (SQUARE)**
   - https://square.itu.dk/student-projects/
   - Reinforcement learning, testing, program analysis, modeling, for robots, probabilistic programs, statistics, reinforcement learning; (and other combinations of these terms)
   - We have robots to play with
2. **In Programming, Logic and Semantics (PLS)**
   - Talk to Patrick Bahr, Marco Carbone, Rasmus Møgelberg, Jesper Bengtsson

Exam

Grades 2023


Grade distribution
30.00%
25.00%
20.00%
15.00%
10.00%
5.00%
0.00%
persons
1.04%
26.04%
15.63%
17.71%
11.46%
15.63%
12.50%
-3
0
2
4
7
10
12
Grade

Basic Facts

- **Date:** Mon, 6 Jan, 09:00 » 13:00
- **Format:** Written exam, 4 hours, on PCs
- **Aids:** All written and online materials
  (code, notes, books, book's GitHub website, etc.)
- **Language Models Not Allowed**. Including ChatGTP and Google Copilot. You risk severe consequences, if discovered.
- **No time to Google a lot,** but recalling details of a type, which you know where to find, will be entirely possible.
- **Independent:** Interaction with other students and other individuals during the exam is not allowed, and will have severe consequences

Basic Facts

- **Questions:** You will get questions in a scala file on LearnIT. No tests
- **Empty bundle:** github.itu.dk, read the instructions before the exam
- You can use any functions and types from the course (textbook, exercises) in your solutions, unless stated otherwise
- Answers to questions in English should be put in comments
- Do not reorder, or re-factor the file.
- **Code must compile.** You fail if it does not
- **Hand-in** a scala file
- **Grading:** A test suite + reading by examiners. Manual grading

Preparation

- (Re-)*reading* the book is not sufficient
- (Re-)doing *exercises*, quizzes, the fake exam is more effective
- *Reflecting on different parts of material* and inventing questions that *combine* them is better
- *Discussing* such questions in a study group is most effective.
- We monitor discord for questions until the exam date (best effort)

Questions?