Software-Engineering Seminar WG PL, WS25 : Functional Pearls

WG Programming Languages

2026-10-22

Supervisors

- Cass Alexandru
- Alexander Dinges
- Felix Winkler

Goals

- Familiarize yourself with a “Functional Pearl”
- Understand it in enough depth to be able to explain it to your peers
- Gain experience in:
  - literature research
  - supervision meetings
  - academic writing
  - peer review
  - making and presenting slides

Functional Pearls

- An elegant solution (an algorithm, design pattern, or datastructure) to an irritating (class of) problem(s)
- Illustrates important programming and modelling techniques, s.a. program/datastructure calculation

Tasks

- Teams of two participants
- Pick a Pearl, then:
- understand it and find your own way to explain the problem it solves and how it does so; this may involve implementing it in your functional programming language of choice
- Explore the publication landscape for later works that reference it, and pick one (MA: two) to investigate
- Write a report [...]
- Make slides and give a presentation [...]

Report

- Language: English (Bachelor: may be in German)
- Use LaTeX or Typst
- If you have taken VFP: It would be extremely nice to formalize the program calculations that are done in some of the pearls using equational reasoning!
- 8-10 pages (Bachelor: 5-8 pages), both team members contribute equally

Presentation

- 25 minutes (~12 minutes per team member)
- **No speaker notes allowed**
- Maximum 100 words per slide
- Maximum 30 slides
- 5 minutes discussion and questions (be prepared to answer questions!)
- ask questions to your peers when they present and give feedback!

Supervision

- Plan your first meeting early, meet regularly
- Come well prepared to the meeting:
  - Take notes of what your supervisor says
  - Make sure you complete ToDos they gave you last time
  - Prepare progress report and questions
- You can submit draft versions of your report to your supervisor intermittently, to request feedback.
- Use the tool `latexdiff` so so your suprevisor can immediately see what you have done with their feedback from the previous round.

Schedule

- First draft of report: 11.12.2026. Hard deadline, as else you're eating into reviewer's time.
- Each of you will be asigned *two* of your peers' draft reports to review. This review will be single-blind.
- Deadline for your reviews is 04.01.2027
- Presentations will be in calendar week 3 or 4, we don't have a room yet
- Final report submission deadline: 05.02.2027

Submissions: As source code + pdfs (+ latexdiff) by email to your supervisor

Presentations Day

- In the theme of a 1-day conference
- There will be sessions of 3 talks, with a coffee break (coffee, tea & biscuits will be provided!) & lunch break in between
- There will be a certificate for “best talk”

Pearls

Supervised by:

- Felix:
  - "Drawing trees"
  - "Deriving tidy drawings of trees"
  - "On building trees with minimum height"
- Cass:
  - "On generating unique names"
  - "Three algorithms on Braun trees"
  - "The Zipper"
- Alex:
  - "On removing duplicates"
  - "The Third Homomorphism Theorem"
  - "Deduction for functional programmers"

Break: 15 mins: Pick your Paper & Partner

After the break: Academic Writing

Academic Writing

Why writing is essential (Prof. Bieniusa)

- It forces you to formulate and clarify your thoughts.
- Allows to develop complex arguments and pin them down.
- It generates transferable knowledge.
- Written material makes your work, ideas, thought process, findings accessible to others $\Leftarrow$ Feedback and evaluation!

Some advice

- Style: Many style guides out there, you're better off waiting for your supervisor to proofread a draft than to read all possible opinionated guides.
- Structure: One paragraph, one message: the introductory sentence should be a "spoiler"
  - A good way to structure your report: As bullet points that you then reformulate as introductory sentences and flesh out into paragraphs
- Within a paragraph: Flow: There shouldn't be jumps between sentences; it should be clear how every sentence is connected to the previous one.

Structuring your report

- Introduction
- Problem Statement
- Contributions & key idea (of the Pearl)
  - You *will* need to use unintroduced concepts $\Rightarrow$ forward references to where you introduce them, as well as for each contribution
- Technical section (reexplain the Pearl, may need sub-sections)
- Section(s) on later work (these may need their own sub-sections)
- Conclusion
- (Bibliography)

Further Reading / Watching

- “How to Write Papers So People Can Read Them” - Derek Dreyer
- “State the problem before describing the solution” - Leslie Lamport
- “How to give a great research talk” - Simon Peyton Jones
- “How to give a talk” - Niels van der Weide

Practical Advice

Finding Literature

- Set up *Zotero*, a Literature Management System
  - In particular, install the Zotero plugin for your browser
- Search for literature on Google Scholar, if unsuccessful, try a general web search
  - Sometimes with older articles it may be easier to find the whole issue of a journal an article appeared in than to find the article itself
  - Add the article via the Zotero plugin from the publisher's page
    - within the uni network you should have access to most articles (else consult the librarian of the CS dept library)
  - Use the "BetterBibTeX" plugin and export Zotero's metadata for the items in your collection to a `.bib` file
- Use Google Scholar's reverse citation ("Cited by") feature to find works that reference your pearl

On Using LLMs

- LLMs will produce garbage scientific writing. They will generate text that:
  1. Uses terminology that *you* don't understand, which you haven't referenced, introduced, which you may not need and which you will be unable to explain or explain the relevance of
  2. repeats definitions and paragraphs with slightly different wording
  3. may contain factually incorrect claims
  4. sounds like sleazy corporate language or a sales pitch
- If you use an LLM, you are claiming authorship of its input, and thus assuming responsibility for it. For the reasons mentioned above, this is an extremely bad idea, and will likely lead to you failing this course.

On Using LLMs

- Only ever use an LLM to generate output you can immediately and indubitably check the correctness of
- In the context of this seminar, this means its utility is essentially reduced to typesetting assistance
- Using LLMs to generate *any* prose in your report, presentation, or reviews of your fellows students' drafts is *forbidden*

How to Pass the Seminar

- Original (= written by you!) text
- Stick to deadlines
- Attend final presentations
- Well written report
  - Conveys the key features
  - Comprehensible English
- Good presentation
  - Comprehensible story
  - Sticks to time limit
  - Able to answer questions
  - Not too much text on the slides
- Meet with your supervisor
- Use a spellchecker!