Joint Master’s Programme in Software Engineering
Introduction for new master students

$UIB - II - PUT$

Magne Haveraaen
University of Bergen
Contents

- Joint Master’s Programme in Software Engineering
- PUT
- About thesis
- About classes
Joint Master’s Programme in Software Engineering

- Supervisors: HVL or UiB or joint, possibly also external
- Compulsory courses
  INF234(A), DAT250(A),
  INF222(S), DAT251(S)
- Two master thesis variants (talk to supervisor)
  - Long: 60 stp, 2 semesters, deadline 1 June 2021 – and 6 courses
    Research/development oriented, needed for PhD
  - Short: 30 stp, 1 semester, strict start/end dates – and 9 courses
- Non-compulsory courses freely selected from HVL / UiB
  **Tip:** do relevant electives first
What is “programutviklingsteori?”

- Programutvikling ≈ software engineering, that is, building and maintaining software systems using experience, tools, processes, common sense and a tiny bit of theory.
- Our research at PUT/II/UiB aims to reverse this order:
  1. given a solid theoretical base for programming, tools, and languages,
  2. build more reliable/correct/faster software systems more effectively
- **Software engineering challenge:**
  how to build and maintain systems that are too *large and complex* for the *human mind* to handle (without help).
What do we do wrong?
PUT — Main Directions

- **Theoretical computer science** — logic, formal methods, type theory, category theory
- **Programming language theory** — programming models, program analysis and transformation, compiler construction, language design, types, generic programming, syntax/grammars, language semantics, domain-specific languages, GUIs
- **Software engineering** — design, construction, testing, development tools, security, language evolution, GUIs

Theory, and applying theory to build real systems
Investigate PL concepts – input to mainstream language evolution

- Fortran standardisation ISO SC22 WG5, ANSI PL22.3 (J3)
  - Generics derived from Magnolia
- C++ standardisation ISO SC22 WG21, ANSI PL22.16
  - Lambdas, Variadics, Concepts
- New Python array library for HPC and ML

Long term research projects

- Magnolia: integrated programming and specification language: Generic programming, Precondition validation, Array algebra
- Multiway dataflow constraint systems: application to GUIs
- Data dependency algebra: parallel programming model

High integrity software systems: safe and secure software

Novel mechanisms:
  compiler construction, software structuring and reuse

Supervisors: Magne, Jaakko, Anya
Thesis Examples from Recent Past

- Evaluation of Property Models for Web GUIs
- A Super User Interface
- C♭: An Intermediate Representation Language for the Purpose of Software Migration to Java and C♯
- What a Machine Can Learn About Code Quality
- A domain-specific dialect for financial-economic calculations using reactive programming
- Program Transformations in Magnolia
- GPU Programming in Magnolia
- Making Software Refactorings Safer

Recall choice of long (60stp) and short (30stp) thesis
About classes

- Consider taking interesting electives early, mandatory classes later!
- Some electives this autumn
  - INF 214 Concurrent programming (Jaakko)
  - INF 220 Program specification (Magne)
  - INF 226 Software security (Håkon)
  - INF 329 Selected Topics in Programming Theory (Marc – irregular)
- Some electives in the spring
  - INF 112 Software Engineering (Siv – undergraduate)
  - INF 222 Programming Languages (Jaakko)
  - INF 223 Category Theory (Uwe)
  - INF 227 Introduction to logic (Michał)
- Some irregular electives
  - INF 210 Modelling of Computing
  - INF 225 Introduction to Program Translation
  - INF 328 Elements of Programming Languages
put.ii.uib.no

Consider engaging via INF219 or INF319

Talk, email, get in touch!

Startup meeting at HVL room D111 – Friday 2019-08-16

Remember the boat trip – Thursday 2019-08-22