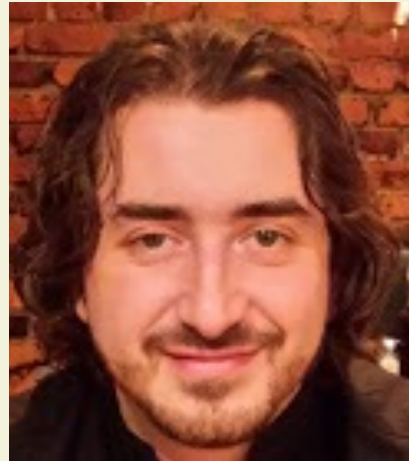


# Master theses available. Three research groups:

Computing Education/Teaching IT (ITU)

Software Engineering (SE)

Programming Technology (PT)



## Software Engineering group

- Located at the 10th floor

# Software Engineering Group

- Contact
  - Viktoria Stray – [stray@ifi.uio.no](mailto:stray@ifi.uio.no)  
(group leader)
  - Yngve Lindsjørn – [ynglin@ifi.uio.no](mailto:ynglin@ifi.uio.no)
  - Antonio Martini – [antonima@ifi.uio.no](mailto:antonima@ifi.uio.no)
  - Gunnar Bergersen –  
[gunnab@ifi.uio.no](mailto:gunnab@ifi.uio.no)
  - Marthe Berntzen –  
[marthenb@ifi.uio.no](mailto:marthenb@ifi.uio.no)
  - Eva Hadler Vihovde –  
[evav@oslomet.no](mailto:evav@oslomet.no)
  - Ketil Stølen – [Ketil.Stolen@sintef.no](mailto:Ketil.Stolen@sintef.no)

## A good idea

Send max 1 page with a proposal for the Master thesis:

- Topic
- Motivation
- Research Question

In addition, if you have a case (company), even better

# Some examples of earlier master thesis in the SE group

- **Challenges and Solutions in Global Agile Software Development**
  - *Investigating Communication, Trust and Knowledge Sharing*
- **Teamwork Quality in a Software Engineering Capstone Course**
  - *A case study of teamwork in a Software Engineering Capstone course during the COVID-19 pandemic*
- **BizDev Teams in Agile Software Development**
  - *A case study of coordination between business and development*
- **Agile BI**
  - *How Norwegian organizations achieve agility in Business Intelligence*
- **Outsourcing in Autonomous Agile BizDevOps Teams**
  - *A case study of globally distributed teams*
- **Fordeler og utfordringer med kryssfunksjonelle produktteam i agil systemutviklingspraksis**
  - *En utforskende casestudie fra en stor nordisk forsikringsaktør med evaluering i lys av prosessforbedringslitteratur*

# Viktoria Stray- 1

- Flere problemstillinger mulig. For eksempel
  - Hvordan sikre en god arbeidsdag i et smidig utviklingsprosjekt?
    - Hvordan skal produktteam organiseres og hvordan jobber de best?
    - Hvordan brukes generativ AI av utviklere? Muligheter og farer.
    - Hvordan sikre autonome effektive smidige team?
    - Hvordan jobbe smart og få til en god digital smidig transformasjon?
  - Samarbeid med bedrifter og SINTEF i et forskningsprosjekt støttet av forskningsrådet  
<https://www.knowit.no/forskning/transformit/>
  - Oppgaven krever at man snakker skandinavisk da man skal gjøre intervjuer og observasjoner i en norsk bedrift

<https://www.mn.uio.no/ifi/personer/vit/stray/>



ITERATE

knowit



# Viktoria Stray- 2

- Systematic literature review
  - Of a selected topic in Software Engineering. For example,
    - Use of Generative AI
    - Use of theories
    - Meetings
  - Can be short or long thesis
    - What is a systematic literature review?
    - <https://libguides.csu.edu.au/review/Systematic>

<https://www.mn.uio.no/ifi/personer/vit/stray/>

## Marthe Berntzen – [Marthenb@ifi.uio.no](mailto:Marthenb@ifi.uio.no)



Flere aktuelle problemstillinger, for eksempel:

- Hvordan lykkes med samarbeid på tvers av smidige utviklingsteam?
- Hvilke hindringer står i veien for effektiv koordinering i store utviklingsbedrifter?
- Hvordan kan roller med ulik kompetanse og bakgrunn samarbeide godt i større utviklingssettinger?
  - Problemstillingene forutsetter at man snakker skandinavisk fordi datainnsamling (intervjuer, observasjon) vil gjennomføres i bedrifter der norsk er arbeidsspråk

*It is also possible to do a [systematic literature review](#) on a SE-related topic, e.g.:*

- *Coordination mechanisms in agile research*
- *Use of theories in the SE literature on coordination (or another related topic)*

<https://www.mn.uio.no/ifi/personer/vit/marthenb/>



# MASTER TOPICS – SOFTWARE ENGINEERING AND TEAMWORK

Yngve Lindsjörn  
[ynglin@ifi.uio.no](mailto:ynglin@ifi.uio.no)  
91549139

- 1) Large-Scale Agile Software Development. Agile methods were first used in small projects with little criticality. How can agile practices be adapted and combined with traditional practices to function effectively in large-scale development and multi-team environment?

Challenges:

- System architecture across teams
- Working agile in “non-agile” organizations and settings
- Consistency across teams for the development practices

~~Inter team dependences~~

---

- Product owner role
- Handling requirements in distributed development
- Managing Technical Debt

- 2) Conduct a literature review of research on teamwork and the relation to team performance and personal success in software development
- 3) Conduct an empirical study (qualitative and/or quantitative) on how teamwork factors such as team leadership, team cohesion, communication and self management effect team performance and personal success in software development teams.

# Master theses

AI, data-driven, software and process quality, design science research

- Check the available theses at <https://www.mn.uio.no/ifi/studier/masteroppgaver/software-engineering/>
- Other topics are possible, you can propose your own!
- Check previous very good theses (A-B grades) at <https://www.mn.uio.no/ifi/english/people/aca/antonima/examples-of-past-master-theses/>
- Contact
  - Antonio Martini @ Software Engineering group
  - [antonima@ifi.uio.no](mailto:antonima@ifi.uio.no)

# Topics

Antonio Martini @ Software Engineering group

- A chatty Digital Twin for factories
  - Modernizing information retrieval and data-driven decision making in factories
  - Combining:
    - Chatbot AI cutting edge technology (mostly Large Language Models)
    - Digital twin communication (how data about an entity, e.g. A company, is organized)
  - Method: Design Science Research
- The hidden debt in bioinformatics software: challenges and consequences of (poor) software quality
  - Technical Debt is a challenge for "traditional" commercial software
    - Average 35% waste
  - What is the situation for bioinformatics software?
    - Field is growing but many programs and projects are developed by non-trained software engineers
  - Method: Survey or interviews with bioinformaticians
- Analytics to support software development (see other theses online)
  - Development of data-driven insights
    - Visualizations, metrics, etc.
  - Method: quantitative, design science research, possibly AI

# Dag Sjøberg og Gunnar Bergersen



## How Generally Useful are Specific Software Tools & Methods?

- **Objective**
  - Investigate: How various tools or methods perform across different settings
  - Understand: Which underlying factors contribute to the success or limitations of these tools/methods
- **Activities**
  - Literature Review: Examine articles that report studies on the impact of existing tools and methods
  - Scripting & Data Handling: Refine existing Python scripts, create new ones, and collect and visualise statistics
  - Comparative Analysis: Compare your findings with results published in two previously published master theses at Ifi and some articles
- **Benefits & Outcomes**
  - Comprehensive understanding of software tool/method evaluation
  - Enhanced Python scripting abilities, and data interpretation and analytical skills



**E-post**

[omidmi@ifi.uio.no](mailto:omidmi@ifi.uio.no)

**Telefon**

+47 22857700

**Mobiltelefon**

+47 45050045

# Exploring the future of Digital Education

- **AI in education**

- *Balance between tech advancements and ethical challenges*
- *Ensuring privacy, transparency and avoiding biases*
- *Blockchain, Big Data, Neural Networks, Adaptive learning*

- **Power of feedback**

- *Beyond binary; AI-driven insights and recommendations*
- *Tailored and continuous course refinement*

- **Beyond traditional exams**

- *Assessing true learning & psychological impacts*

- **Peer Tutoring, Accessibility and Inclusivity, Gamification, Collaboration**

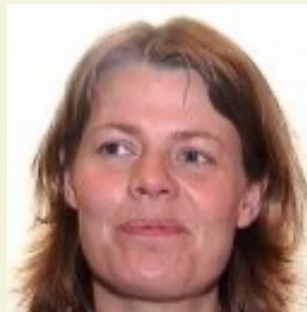
# Computing Education (ITU) master theses 2023

Odd Petter Sand



oddps@ifi.uio.no

Siri Moe Jensen



siriamj@ifi.uio.no

Ragnhild Kobro Runde



ragnhilk@ifi.uio.no

# Five possible topics

- Abstraction in programming education
- Programming students struggling to understand the machine side
- Students' understanding in different introductory programming courses
- Analyzing results from IN1000 (introduction to programming)
- Programming in schools



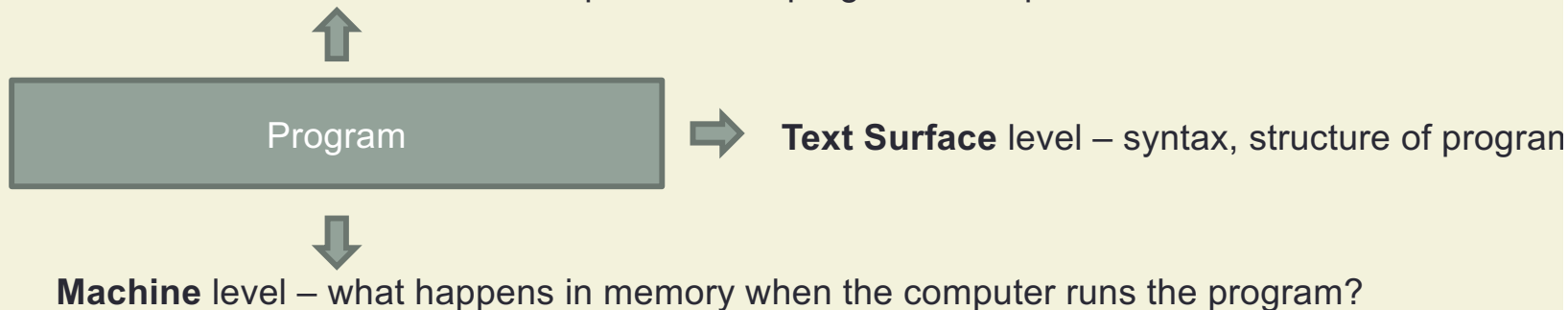
# Abstraction in programming education

- Abstraction: “zooming in/out”, or “separating the **what** (interface) from the **how** (implementation)”
- Possible topics:
  - In what different ways do students approach tasks that require abstraction (and how do we classify such tasks)?
  - How do students experience built-in abstraction mechanisms in Python (functions, classes...)
  - Compare contexts: IN1000/IN1900/BIOS1100
- Odd Petter Sand & Ragnhild Kobro Runde
- Need to speak/understand Norwegian



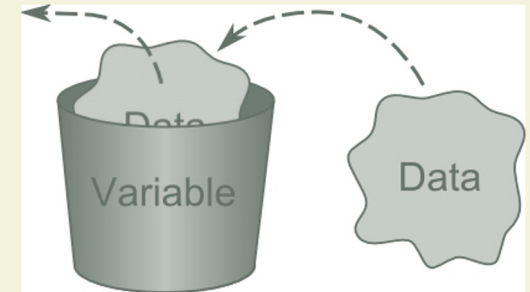
# Programming students struggling to understand the machine side

**Functional** level – what is the real-life problem the program attempts to solve?



- Hypothesis: Some students have NO machine understanding
- Possible questions:
  - How to identify students with no machine understanding?
  - What are the thought processes of these students?
  - How to help the students get an understanding of the machine side?
- Supervisors: Odd Petter Sand and Ragnhild Kobro Runde
- Language requirement: Norwegian and English

# Students' understanding in different introductory programming courses



- How does the academic context influence students' learning and understanding of programming?
  - How are programming taught in different courses?
  - What kind of examples are used, and what programming concepts are emphasized?
- Possible frameworks for analysis:
  - The block model
  - Variable roles
- Supervisors: Ragnhild Kobro Runde and Siri Moe Jensen
- Language requirement: Norwegian

# Analyzing results from IN1000 (introduction to programming)



- One or more theses analyzing new and historical data in order to answer questions like:
  - How do individual students' understanding and skills develop throughout the semester?
  - What can explain differences in exam results for different programmes, and what measures can possibly be taken?
  - How well do scores on automatic graded tasks correlate with scores on more creative exam tasks?
- Supervisors: Siri Moe Jensen and Ragnhild Kobro Runde
- Language requirement: Basic understanding of Norwegian

# Programmering i skolen

- Er du interessert i programmering i skolen?
  - Kjenner du lærere som underviser matematikk, programmering eller informasjonsteknologi på ungdomsskolen eller videregående skole og som kunne tenke seg å delta i et forskningsprosjekt sammen med en eller flere av klassene sine?
- Ta kontakt med Ragnhild Kobro Runde for diskusjon om mulige temaer for masteroppgaver.

# Interested?

## Contact us for further discussions

Odd Petter Sand



[oddps@ifi.uio.no](mailto:oddps@ifi.uio.no)

Siri Moe Jensen



[siriamj@ifi.uio.no](mailto:siriamj@ifi.uio.no)

Ragnhild Kobro Runde



[ragnhilk@ifi.uio.no](mailto:ragnhilk@ifi.uio.no)



**UiO : Department of Informatics**  
University of Oslo

## **UiO Master Thesis proposals, Fall 2023**

Michael Kirkedal Thomsen  
[michakt@ifi.uio.no](mailto:michakt@ifi.uio.no)

Oct 10 2023

# Areas of interest

- Domain specific programming languages and computation models in general
- Information theory and computation
- Decentral systems (information handling and consensus)
- Models for improved reliability; e.g. reversibility, probabilistic and quantum computations

## Understanding Programming Languages

- DSLs are everywhere
- Problems in PL are often uncomputable
- Understand a model of the problem
- Implemented your own programming languages



# Trust language for decentralised maritime systems

- Learn about maritime communication
- Project in collaboration with industry partners (e.g. P3KI, Germany)
- Understand de-centralised systems design
- Understand modeling provably of secure relations
- Implement domain specific language, focus on
  - Aids-to-Navigation
  - Navigational warnings
  - Information broadcast





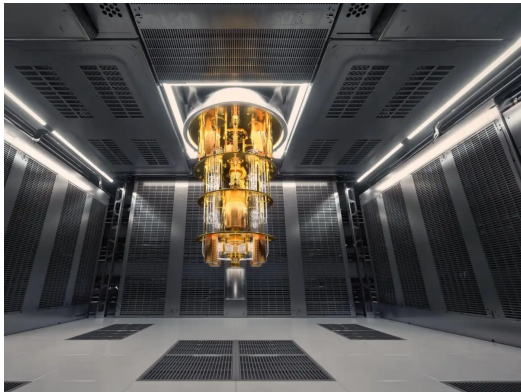
# Energy consumption of programs and computation

- Measure energy consumption of existing programming language
- Understand possible contributions
  - Trade-offs
  - Program libraries
  - Compiler effects
- Understanding of models for energy-aware programming
- Implement needed tools



# Programming languages with quantum effects

- Learn the basics of quantum computations
- Understand quantum effects
- Understand how you program quantum computers
  - Know the limitations
- Design programming languages with quantum effects
- Implement compilers
- Run on real quantum computers





**UiO** : **Department of Informatics**  
University of Oslo



**Michael Kirkedal Thomsen**



**UiO Master Thesis proposals, Fall  
2023**





# MSc Theses Proposals by Paulo Ferreira



**Learn by doing !**

- Professor at UiO – office in room 10460
  - <https://www.mn.uio.no/ifi/english/people/aca/paulofe/index.html>
- MSc theses will be done at UiO / PT (10<sup>th</sup> floor)
- Do you have your own suggestions? Let's talk !
- More information:
  - come to room 10460 and we have a chat
  - send me an email: [paulofe@ifi.uio.no](mailto:paulofe@ifi.uio.no)
  - Zoom link: <https://uio.zoom.us/j/8253296061>
  - contact me via Skype, Viber, WhatsApp, etc...
- Requirements:
  - good tracking record (grades, courses), enthusiasm, and commitment.
- MSc themes:
  - Fog and Cloud Computing
  - Java Virtual Machine/Android
  - Ubiquitous/Mobile Systems
  - Distributed Systems

# edgeTrans – Visual Simulation of Cloud, Cloudlets, and Sensors

- **Background:**

- An existing simulator for encounter based mobile apps is available (<https://rodrigo-bruno.github.io/mentoring/81914-fernando-moreira-dissertacao.pdf>)
- This emulator does not support several options related to fog computing (e.g., cloud data centers, sensors, openstreetmap, etc.)
- Thus, the emulator must be changed accordingly

- **Goal:**

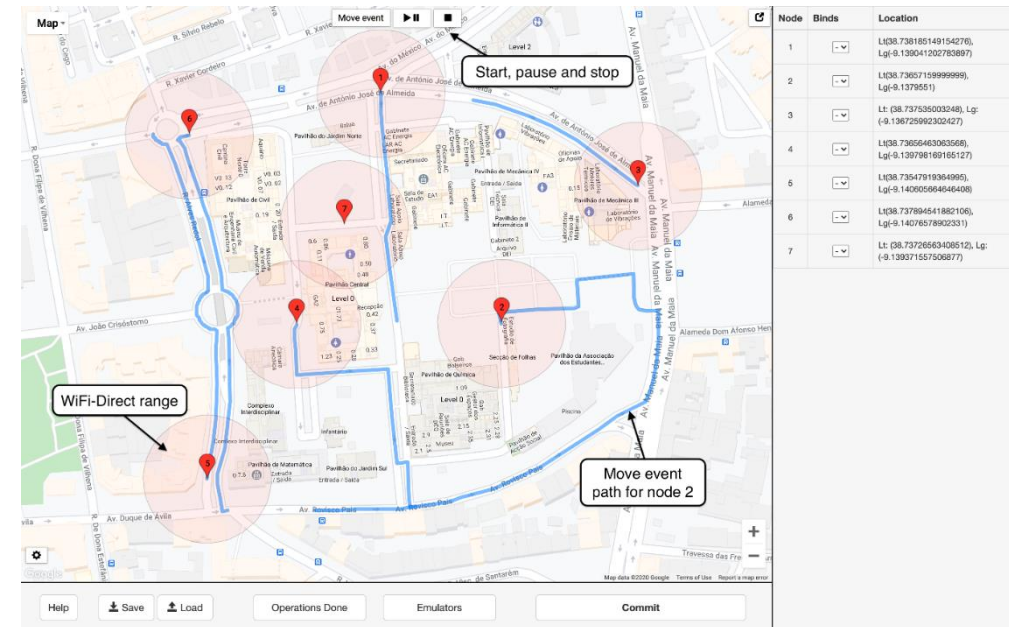
- Create/install the emulators for other devices
- Start with Raspberry PI and move on to a cloud data center based on Linux

- **Requirements:**

- The candidate must enjoy and have adequate skills to deal with several programming languages, and mobile system issues implementation
- The candidate should be able to program in Java
- Also relevant is a good tracking record (grades, classes done), enthusiasm, and commitment

- **Supervisors:**

- Paulo Ferreira and Lyla Vijouyeh



# mapDetect –Automatic Detection of the Transport Mode with a Map

- **Background:**

- Automatic detection of a transport mode is usually done with a ML algorithm
- A new version of Woorti (<https://www.woorti.com/>), called FogTMDetector, is a local transport mode detector, i.e., it runs only on the smartphone not requiring network connection, and uses a classifier based on the Random-Forest ([https://www.youtube.com/watch?v=o2E1md1t69U&ab\\_channel=TheMoTiVProject](https://www.youtube.com/watch?v=o2E1md1t69U&ab_channel=TheMoTiVProject))
- However, sometimes Woorti/FogTMDetector, is not capable of making a correct detection of the transport mode

- **Goal:**

- Improve the current Woorti/FogTMDetector, solution, with a map
- E.g., based on the lines used by public transports, better identify the transport mode being used
- The information must be minimized so that the any smartphone can use it and it does not occupy much memory

- **Requirements:**

- The candidate should be able to program in Java
- The candidate must enjoy and have adequate skills to deal in Java, Android and mobile issues implementation
- Also relevant is a good tracking record (grades, classes done), enthusiasm, and commitment

- **Supervisors:**

- Paulo Ferreira



# Energy Aware Garbage Collection

- **Background:**

- There is a lack of information regarding how much energy is spent by a GC (Garbage Collector)
- Energy spent by a GC (as in other software) is becoming more important day after day
- This means that sustainability is of great concern being the energy spent a crucial aspect

- **Goal:**

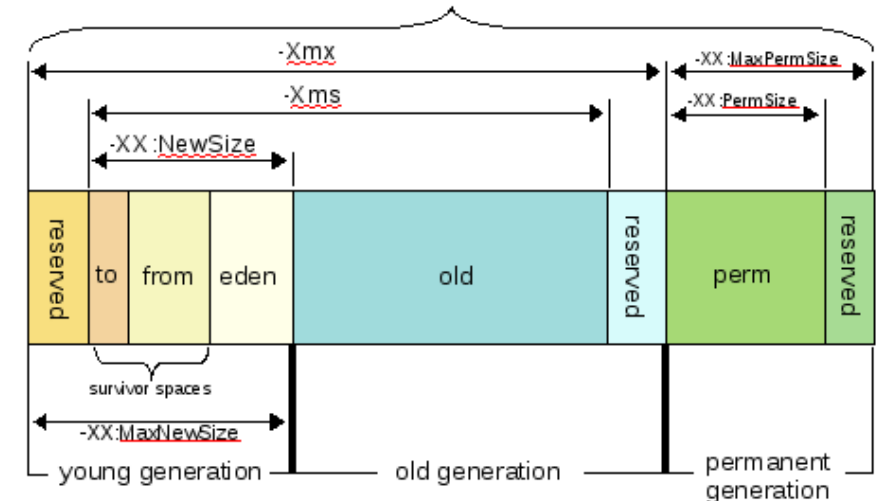
- Perform a related work search of similar energy aware GCs
- Study the behaviour of production GCs energy-wise (Parallel, GC1, Shenandoah, ZGC)
- Propose solutions to make a GC to better use the energy available (e.g., run slowly when there is no urgency, save cores when time is available, etc.)

- **Requirements:**

- The candidate must enjoy and have adequate skills to deal with systems' implementation
- The candidate should be able to program in Java
- Also relevant is a good tracking record (grades, classes done), enthusiasm, and commitment

- **Supervisors:**

- Paulo Ferreira and Eric Jul





# GCgraalVM – Garbage Collection for the Graal Virtual Machine

- **Background:**

- Currently, the NG2C Garbage Collector uses the ROLP system (running in the OpenJDK JVM) to detect the age of created objects so that the NG2C allocates such objects close to each other
- Such allocation contributes to improve GC latency while keeping a good throughput

- **Goal:**

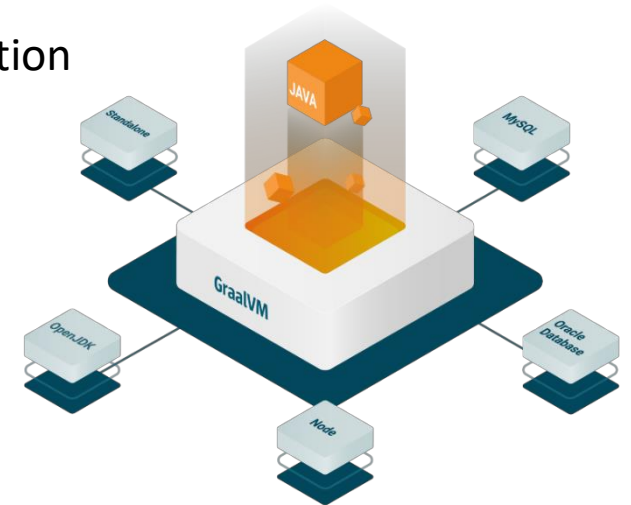
- Develop ROLP for the GraalVM (it is a innovative Virtual Machine that supports several languages - Java, JS, Ruby, etc.)
- The development should be language agnostic

- **Requirements:**

- The candidate must enjoy and have adequate skills to deal with systems' implementation
- The candidate should be able to program in C++ and Java
- Also relevant is a good tracking record (grades, classes done), enthusiasm, and commitment

- **Supervisors:**

- Paulo Ferreira



# MSc Theses Proposals by Paulo Ferreira

- Questions?
  - let's talk
  - feel free to come to room 10460, or
  - send me an email: [paulofe@uio.no](mailto:paulofe@uio.no) or
  - Zoom link: <https://uio.zoom.us/j/8253296061> or
  - contact me via Skype, Viber, WhatsApp, etc...



[web page \(these slides\): https://www.mn.uio.no/ifi/english/people/aca/paulofe/index.html](https://www.mn.uio.no/ifi/english/people/aca/paulofe/index.html)

Courses at UiO :

- Fog and Cloud Computing: <https://www.uio.no/studier/emner/matnat/ifi/IN5700/index-eng.html> (IN5700)
- Programming Ubiquitous Things: <https://www.uio.no/studier/emner/matnat/ifi/IN5600/index-eng.html> (IN5600)



# Master's Thesis Proposals

Professor Eric Jul  
Programming Technology Group  
[ericbj@ifi.uio.no](mailto:ericbj@ifi.uio.no)  
Fall 2023



I teach **IN5570 Distributed Objects** & **IN3030/IN4330 Efficient Parallel Programming**

Veiledning: enten på **dansk** or **English (American)**

Strong believer in ***Learning-by-doing*** so project are much about ***hands-on programming***  
Projects:

- Power efficient programming in several areas:
  - Parallel Computing for Multi-Core
  - Android power consumption
  - Power efficiency of different programming languages
- Distributed Objects
- Flight Mode detection using Android sensors



# Energy Efficient Parallel Computing for Multi-Core Systems

Projects that involve making «kick-ass» efficient parallel programs that existing algorithms and make them run **POWER EFFICIENT** on multi-core systems.

Investigate how to monitor the energy efficiency of a computation.  
The PT Group has hardware that can be used for such monitoring.

Requires having taken IN3030 OR taking IN4330/Spring 2024 😊



# Energy Efficiency of Programming Languages

Investigate how to monitor the energy efficiency of different programming languages.

The PT Group has hardware that can be used for such monitoring.



## Energy Efficient Computing for Android

**Investigate how to monitor the energy efficiency of a computation on Android.**

**Find out what uses the power and how much different components use, e.g., screen, network, CPU, etc.**



## Self-Adaptive Computing using Multi-Core Systems

Many concurrent programs try to utilize all the cores present on the processor that they run on.

Often the number of cores can be obtained from the underlying run-time system, but not always – and are we getting the true answer?

Furthermore, cache-friendliness is also important for programs that handle large amounts of data.

Therefore it is important to know about the cache structure of the processor and its attached memory.

These parameters are not always easy to obtain – and even so, are we getting the true answer?

.

**Challenge:** *Write a program in Java that experimentally tries to find all these parameters! Then use it to write «self-adaptive» programs.*



## Distributed Objects – Language Mechanisms

- **Implementing the TRACK mechanism in Emerald**
  - TRACK is a proposed language construct that allows an object, A, to keep track of where another object is currently located in a distributed system.
  - Combination of implementation (both compiler and runtime) and language level usage of the concept.
  - Co-author an article with me on the subject for a conference
  
- **Type-based Access Control, Locks, and Authentication**
  - Using the restrict-to mechanism to limit access control
  - Implement a number of examples of how to provide language-based control mechanism
  - Co-author an article with me on the subject for a conference





## Refining Techniques for Automatic Flight Mode Detection

An earlier project, flyDetect, has developed tools for automatic detection of flight mode for a smartphone based on acceleration and barometric pressure.

The project had good results, but its approach for landing detection is vulnerable to false positives because of certain altitude changes.  
cabin.

The proposed project would investigate new approaches for detecting landing as to make the detection more robust.

Furthermore, adapting to specific types of aircraft, may further improve the results.