

Course Essentials

Course title and number	INF 222 – Programming Languages
Term	Spring 2017
Course pages	http://www.iit.uib.no/~jjarvi/INF222/
Instructor	Jaakko Järvi
Email address	jaakko.jarvi@it.uib.no

Course Description

The course gives a practical and theoretical introduction to the semantics of programming language constructs. The course covers a range from micro level constructs, e.g., parameter passing, to macro level constructs, e.g., module systems. Reflecting on the influence of the constructs on software quality forms part of the course.

We discuss the notions of different approaches to types and type checking in languages. Some features we study by using them in programs, some by implementing them in small interpreters in Haskell. Occasionally we will work on other languages as well.

Learning Outcomes

After successfully completing the course, the students shall be able to:

- Define micro level constructs by programming evaluators.
- Define macro level constructs and how they are emulated in common programming languages.
- Identify language features that benefit software integrity (reliability, robustness, and security) and software validation (verification and testing).

The course will help you to become more productive with the languages of today and more prepared to learn (perhaps also implement) new programming languages in the future.

Textbook and Resource Material

Material from the following books will be used:

- Ralf Lämmel: [Software Languages \(Draft\)](#) This book is not publicly available yet. (Electronic) copies will be provided for course attendees.
- William Cook: [Anatomy of Programming Languages \(Draft\)](#)

Course Topics

- Basic notions of programming languages
 - Representing programs in a host programming language
 - Phases in language processing

- Language processing
 - Abstract and concrete grammar, parsing, pretty printing
 - Operational semantics of programming languages
 - Interpretation
 - Types and type checking
 - Lambda Calculus
- Important abstractions/concepts/language features
 - Closures, Evaluation contexts, continuations
 - Evaluation order, parameter passing
 - Parametric polymorphism, Existential types
 - Subtype polymorphism and generics
 - Abstractions for concurrent programming

Assignments

The course will have a continuous stream of compulsory assignments, many of them involve programming in Haskell.