CS-320

Computer language processing

Kuncak Viktor

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computer science minor	E	Opt.
Computer science	BA6	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Courses	2 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

We teach the fundamental aspects of analyzing and interpreting computer languages, including the techniques to build compilers. You will build a working compiler from an elegant functional language into machine code using a popular backend called LLVM (https://llvm.org)

Content

See https://lara.epfl.ch/w/cc
1. Overview, source languages and run-time models
2. Review of formal languages
3. Lexical analysis
4. Syntactic analysis (parsing)
5. Name analysis
6. Type checking
7. Code generation
8. Correctness of compilers

Keywords

programming language;
compiler;
interpreter;
regular expression;
context-free grammar;
type system;
code generation;
static code analysis

Learning Prerequisites

Recommended courses

Discrete Mathematics
Theory of computation
Functional Programming
Computer architecture

Learning Outcomes

By the end of the course, the student must be able to:

- Design a programming language
- Construct a compiler
- Coordinate development with project partner
- Formulate correctness conditions for compiler
- Estimate time to implement a programming language feature
- Produce a working programming language implementation
- Decide which language features make implementation difficult
- Specify programming language and compiler functionality

Transversal skills

- Assess progress against the plan, and adapt the plan as appropriate.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Respect the rules of the institution in which you are working.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate a capacity for creativity.
- Take feedback (critique) and respond in an appropriate manner.
- Make an oral presentation.
- Write a scientific or technical report.

Teaching methods

Lectures, exercises, labs

Expected student activities

- Follow lectures
- Project work, indepdently and under supervision of assistants

Assessment methods

The grade is based on a midterm exam (30%) as well as programming, testing, documentation, and presentation of several projects done on student's own laptops during the semester (70%).
Different groups of students may be assigned different variants of projects. There may be small but unavoidable variations in the difficulty of different variants.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Andrew W. Appel, **Modern compiler implementation in Java** (or **ML**), Addison-Wesley 1997 (full PDF available from EPFL library)
Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman: **Compilers: Principles, Techniques, and Tools** (2nd Edition, 2006)

Ressources en bibliothèque

- Modern compiler implementation in Java / Appel

- Compilers, principle, techniques and tools / Aho

Notes/Handbook

http://lara.epfl.ch/w/cc
Faboulous and gently paced videos: https://www.coursera.org/course/compilers

Websites

- https://lara.epfl.ch/w/cc

Moodle Link

- https://go.epfl.ch/CS-320

Prerequisite for

Advanced compiler construction
Recommended for Foundations of software