# Foundations of software

Doeraene Sébastien

<table>
<thead>
<tr>
<th>Cursus</th>
<th>Sem.</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer science</td>
<td>MA1,</td>
<td>Obl.</td>
</tr>
<tr>
<td>Minor</td>
<td>MA3</td>
<td></td>
</tr>
<tr>
<td>Computer science</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MA1, MA3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cybersecurity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MA1, MA3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Language</th>
<th>Credits</th>
<th>Session</th>
<th>Semester</th>
<th>Exam</th>
<th>Workload</th>
<th>Weeks</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>4</td>
<td>Winter</td>
<td>Fall</td>
<td>Written</td>
<td>120h</td>
<td>14</td>
<td>4 weekly</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Summary

The course introduces the foundations on which programs and programming languages are built. It introduces syntax, types and semantics as building blocks that together define the properties of a program part or a language. Students will learn how to apply these concepts in their reasoning.

## Content

- simple types, lambda-calculus
- normalization, references, exceptions
- subtyping
- recursive types
- polymorphism
- advanced features of the Scala type system

## Keywords

lambda-calculus, type systems

## Learning Prerequisites

### Recommended courses

Advanced topics in programming, Compiler construction

### Important concepts to start the course

Functional programming
Basic knowledge of formal languages

## Learning Outcomes

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

- Argue design decisions of programming languages
- Assess / Evaluate soundness of type systems
- Compose higher-order functions
- Verify progress and preservation in type systems
- Work out / Determine operational equivalences
- Carry out projects of 2-3 weeks duration
- Distinguish valid from invalid proofs
- Implement type systems and operational semantics
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.
- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Manage priorities.

Teaching methods

Ex cathedra, practical exercises

Assessment methods

Final written exam during the exam session (60%)
Projects and homeworks by groups during the semester (40%)

Resources

Ressources en bibliothèque

Moodle Link

- https://go.epfl.ch/CS-452