## Module title
Languages and Their Compilers

<table>
<thead>
<tr>
<th>Module code</th>
<th>Level</th>
<th>Hours per week</th>
<th>ECTS credits</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tba</td>
<td>Bachelor (B.Sc.)</td>
<td>4</td>
<td>5</td>
<td>2 weeks block course + virtual lectures</td>
</tr>
</tbody>
</table>

### Module instructor
Dr. Lenka Kosková, Technical University Liberec

### Lecture type
Lectures + Guided Lab Sessions

### Prerequisite(s)
Intermediate Programming Ability

### Grading
Coursework compiler writing exercise

### Description
The course discusses internal mechanisms of compilers and interpreters of programming languages. Basic principles of compilation are explained as well as the structure of a compiler and steps of compilation, including Lexical analysis, parsing techniques, syntactical and semantic analysis.

The course covers deterministic languages, LL a LR languages. It focuses also to compilation of program constructs, code generation and optimization. Toolchains and build automation software are demonstrated in labs.

### Content

**Lectures:**
1. Languages and compilers, internal compiler structure.
2. Build automation tools and toolchains.
3. How the grammar theory meets real language specifications (Java, Python)
4. The lexical analysis
5. The syntactic analysis
6. The semantic analysis
7. Intermediate code
8. The optimization
9. Code generation
10. JIT compilers and interpreters

**Labs:**
1. - 2. The compilation process, compiler directives, macros, and pre-processors (GCC)
3. - 4. The build automation tools (make, Maven, Ant)
5. - 6. Lexical analyzer construction (Flex, ANTLR, JavaCC)
7. - 8. Syntactic analyzer construction (Yacc, ANTLR, JavaCC)
9. - 10. Optimization levels and final product

### Textbook/teaching material

Note: this is not the official course descriptor according to the “Studien- und Prüfungsordnung” (SPO)