**Tárgy neve: Compilers**

**Tárgyfelelős neve: Dévai Gergely**

**Tárgyfelelős tudományos fokozata: PhD**

**Tárgyfelelős MAB szerinti akkreditációs státusza: AR**

**Az oktatás célja angolul / Aim of the subject:**
- **Knowledge**
  - General and specific mathematical and computational principles, facts, rules, contexts, and procedures that form the basis of computer science. The relevant fields are computation theory, the design and analysis of algorithms, formal languages, and automata theory.
  - General theories, contexts, and facts, as well as the related concepts in the field of computer science, in particular: the methodological foundations of programming, programming languages, compilers.
- **Abilities:**
  - Apply general and specific mathematical and computational principles, facts, rules, and contexts in the field of information technology.
  - Apply formal models of computer science.
- **Attitude:**
  - Be open to learning about and embracing professional and technological developments and innovations in their field of competence and specialisation.
- **Autonomy, responsibility:**
  - Strive for efficiency and high quality at work.

**Az oktatás tartalma angolul / Major topics:**
- Introduction, rationale. The general design of compilers.
- Lexical analysis: regular languages, regular expressions, and (non-)deterministic finite automata. Thompson's construction, epsilon-elimination, subset construction. The principles of maximal munch and token priorities.
- Syntactic analysis: context-free languages and context-free grammars, pushdown automata. Sentential forms and handles, backtracking and predictive parsing. Top-down and bottom-up approaches, such as LL(1), SLR(1), LR(1) and LALR(1).
- Semantic analysis: context-sensitive languages and attribute grammars. Synthesised and inherited attributes. Binder analysis and type checking with attribute grammars. Using S-attributed grammars in bottom-up parsing.
- Code generation: compositional generation of code strings in symbol attributes. Assembly and code generation templates for simple imperative statements.
- Practice: defining simple imperative programming languages with regular expressions, context-free grammars, and attribute grammars. Compiler implementation with analyser generators.

**A számonkérés és értékelés rendszere angolul / Requirements and evaluation:**
practical grade & exam

**Irodalom / Literature:**

- Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman. Compilers: Principles, Techniques, and Tools. Addison Wesley, 1985.
- Kenneth C. Louden. Compiler Construction: Principles and Practice. PWS Pub. Co., 1997.
- Torben Ægidius Mogensen. Introduction to Compiler Design. Springer Publishing Company, 2011.
- Grune, D., van Reeuwijk, K., Bal, H.E., Jacobs, C.J.H., Langendoen, K. Modern Compiler Design. Springer-Verlag, 2012.