LUNDS UNIVERSITET
Lunds Tekniska Högskola

Course syllabus

Programspråkskoncept
Concepts of Programming Languages

EDAP05, 7,5 credits, A (Second Cycle)

Valid for: 2023/24
Faculty: Faculty of Engineering, LTH
Decided by: PLED C/D
Date of Decision: 2023-04-18

General Information

Language of instruction: The course will be given in English

Aim

Software engineers encounter different programming languages throughout their career. While each new language brings with it new libraries and marketing terms, the underlying key concepts of these languages are typically selected from a relatively small pool of programming language concepts. This course aims to give students an overview over and familiarity with those common concepts to allow them to better understand and more quickly adapt to new languages. In addition, the course aims to support students who work on the design of (domain-specific) programming languages in getting a broader overview over their design choices.

Learning outcomes

Knowledge and understanding
For a passing grade the student must

- understand formalisms for syntactic and semantic descriptions, including context-free grammars.
- understand basic language concepts: variables, control-flow constructs, type systems, statements, expressions, and side effects.
- understand modular language concepts: modules, classes, separate compilation.
- understand language paradigms: imperative, functional, logical, object-oriented.

Competences and skills

For a passing grade the student must

- be able to exploit different language concepts and paradigms when writing software.
- be able to reason about language concepts by using suitable formal descriptions.

Judgement and approach
For a passing grade the student must

- be able to demonstrate the differences between different language concepts and reason about relative benefits and disadvantages.

Contents

Names, scopes, and bindings. Types, including type checking and type inference. Data structures. Different forms of language expressions, statements, and declarations. Control flow, including exceptional control flow. Trade-offs between expressivity, readability, and resilience to errors. Binding mechanisms. Execution models of programming languages. Subroutines and parameter-passing. Modularisation and separate compilation. Reflection and generative programming. Formal descriptions of syntax and semantics.

Examination details

Grading scale: TH - (U,3,4,5) - (Fail, Three, Four, Five)
Assessment: Written examination. To qualify for the written examination, students must have completed their programming assignments. The final grade for the entire course is based on the result of the written examination.

The examiner, in consultation with Disability Support Services, may deviate from the regular form of examination in order to provide a permanently disabled student with a form of examination equivalent to that of a student without a disability.

Parts
Code: 0119. **Name:** Laboratory Work.
Credits: 3. **Grading scale:** UG. **Assessment:** Completed laboratory exercises. **Contents:** Laboratory work
Code: 0219. **Name:** Written Examination.
Credits: 4,5. **Grading scale:** TH. **Assessment:** Written examination. The final grade of the entire course is based on the result of this exam. To qualify for the written exam, students must have completed their laboratory work. **Contents:** Written examination

Admission

Admission requirements:

- EDAA01 Programming - Second Course or EDAA30 Programming in Java - Second Course

The number of participants is limited to: No

Reading list

- Robert W. Sebesta: Concepts of Programming Languages, Global Edition. Addison Wesley, 2016, ISBN: 1292100559. Textbook.

Contact and other information

Course coordinator: Christoph Reichenbach, christoph.reichenbach@cs.lth.se
Course homepage: http://cs.lth.se/edap05
Further information: Course on hold pending redesign.