

Department and Course Number: CSc 201
Course Coordinator: Cui Zhang, Professor of Computer Science
Course Title: Programming Language Principles
Total Credits: 3 units

Current Catalog Description:

Notations for the specification of programming language syntax and semantics; attribute, translational, operational, axiomatic, algebraic, denotational, and action semantics. Applications of programming language syntax and programming language semantics. Use of meta languages to generate executable language definitions for language implementation, program transformation, program property analysis, and rapid software prototyping. Principles of logic, functional and object-oriented programming languages. **Prerequisite:** Fully classified graduate standing in Computer Science or Software Engineering.

Textbooks:

1. R. A. Finkel, Advanced Programming Language Design, Addison Wesley, 1996.
2. J. D. Ullman, Elements of ML Programming, Second Edition, Prentice Hall, 1998.

References:

1. P. Linz, An Introduction to Formal Languages and Automata, Second Edition, Jones and Bartlett Publishers, 1997.
2. B. Meyer, Introduction to the Theory of Programming Languages, Prentice Hall, 1990/1991 (a new edition is to be published).
3. K. Slonneger and B. L. Kurtz, Formal Syntax and Semantics of Programming Languages.

Course Goals:

1. To understand sufficiently concepts and various notations for formal definitions of programming language syntax and semantics, and their applications including language design and implementation, program transformation, program analysis, and verification.
2. To understand sufficiently the advanced programming language design issues and theoretical background, and to be able to choose an appropriate programming paradigm for any given project.
3. To be able to do meta level programming for the generation of executable specification of programming language syntax and semantics which has applications including software prototyping.

Prerequisites by Topic:

1. Fundamental data structures and algorithms to manipulate them
2. An assembly language
3. A high level programming language
4. Formal languages and their relationship to automata
5. Representation of programming language syntax (e.g., BNF)

Major Topics Covered in the Course:

1. Introduction, Evolution of programming languages (1 hour)
2. Mathematical background (2 hours)
3. Formal languages, Automata, Syntactic processing (3 hours)
4. Advanced programming language design issues (4 hours)
5. Lambda calculus (3 hours)
6. Functional programming, SML, Executable specification and applications (6 hours)
7. Abstract syntax, Denotational semantics (6 hours)
8. Axiomatic semantics (3 hours)
9. Attribute grammar, Translational, operational, algebraic, and action semantics (4 hours)
10. Logic programming languages and applications (3 hours)
11. Object-oriented programming languages (3 hours)
12. Trends in programming languages, Representative programming paradigms and languages (6 hours)
13. Midtem (1 hours)

Laboratory Project:

In addition to homework assignments, two team projects are designed for this course. For the first project, every team of two students will be required to generate, by programming in SML, an executable syntactic and semantic specification for a small programming language. This executable specification can be directly used as a rapid prototype of the language system. The project duration is about 6 weeks. Please see the Oral and Written Communication section for the second project.

Estimate CSAB Category Content:

	Core	Advanced
Data Structure		1.5 hours
Algorithms		1.5 hours
Software Design		3 hours
Computer Organization and Architecture		
Concept of Programming Languages		12 hours

Oral and Written Communication:

When completing the first team project (please see the Laboratory Project section above), each team is required to submit a written report on the first project of executable syntactic and semantic specification for a given programming language. As the second project, under the guidance of the instructor, students are required to select a topic, from the topic list provided by the instructor, to study the latest publications on the advances and representative programming language paradigms and languages. Each team is required to write/prepare notes/viewgraphs on the team study results and give a 25 minutes oral presentation to the class. This project duration is about 4 weeks.

Social and Ethical Issue:

No significant component

Theoretical Content:

Mathematical background (2 hours), formal languages theory and automata (3 hours), Lambda calculus (3 hours), abstract syntax (1 hour), denotational semantics (5 hours), axiomatic semantics (3 hours), other forms of formal semantics (4 hours).

Analysis and Design:

Program specification and property analysis, formal semantic definition and analysis, and executable formal semantic specification as rapid prototype are covered by this course. Please also see the Laboratory Project section above.

Rev. 07/02/03