

CS373 Syllabus  
Spring 2008  
Theory of Programming Languages

**Objectives:**

Students will examine a variety of topics important to the understanding of programming languages, including syntax, semantics, type checking, parameter passing, blocking, scope, control, procedural and data abstraction, and encapsulation. Students will study different classes of languages: procedural, logic, functional, object oriented, and hybrids, together with paradigms related to each. Students will identify criteria for choosing an appropriate language for a particular application. They will study concepts pertinent to language design and implementation, including formal aspects of languages. They will become familiar with choices that force tradeoffs among desirable characteristics of a given language.

**Textbook:** **Programming Language Pragmatics** by Michael Scott

**Instructor:** Joan Krone

Olin 211

587-6484

<http://www.denison.edu/~krone>

Labs, tests, and homework will be announced during class meetings.

Each student will choose from the following major projects:

1. Design an all-purpose programming language.
2. Write a translator for a language.
3. Become a guru in a particular language. A collection of questions to answer about the chosen language is available as a guide.
4. Perform a detailed comparative analysis between two languages or among three or more languages using the questions as a guide.

Students will make periodic class presentations related to their projects.

**Grading**

midterms	25%
final	25%
projects	35%
class contributions	15%

## Schedule

Week	Readings	Topics
Jan. 14	Chapter 1 Chapter 2	Language classification; design and compilation Syntax and grammars
21	Chapter 2	Parsing
28	Chapter 3 Chapter 3	Binding, storage management, scope Referencing environments, separate compilation
Feb. 4	Chapter 4	Semantic analysis
11	Chapter 5 Chapter 6	Target Machine Architecture Control flow
18	Chapter 7	Data types
25	Chapter 8	Subroutines
Mar. 3	Chapter 9	Data abstraction and Object Orientation
17	Chapter 9	Parameter modes, exception handling, coroutines
24	Chapter 10	Alternative Models: functional
31	Chapter 11 Chapter 12	Logic Concurrency
Apr. 7	Chapter 13 Chapter 14	Scripting Languages Language Implementation
14	Project Work	
21	Student Presentations	