CS 310 – Principles of Programming Languages

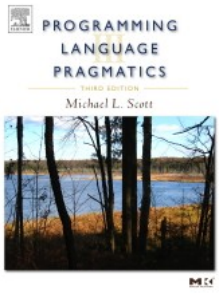
Nicholas Coleman, Ph.D.,
Assistant Professor of Computer Science

Contact Info: office: 442-3272 Engineering Room 101N
ncolema2@mix.wvu.edu http://cs.wvutech.edu/ncoleman

Office Hours : Mon/Wed 9am - 11am, 1pm - 2pm ; Thu/Fri 9am - 11am

Textbook:

Programming Language Pragmatics, 3rd ed.



Michael L. Scott

ISBN: 978-0-12-374514-9
Publisher: Elsevier/
Morgan Kaufmann
Copyright: 2009

Reference: http://www.cs.rochester.edu/u/scott/pragmatics/

Course Homepage: http://cs.wvutech.edu/ncoleman/ppl/

CSCI 310. Principles of Programming Language *3 hrs (*§CS 201*)*

A study of the organization and implementation of programming languages.
Topics include characteristics of grammars, Backus-Naur Form, language specification, data types and structures, the effect of runtime environment on the various features of programming languages, and an introduction to parsing. Students will be required to write simple programs in languages such as Pascal, C/C++, Java, FORTRAN, Ada, Scheme, Python, and Prolog.

Course Goals and Objectives:

- The student will be able to describe the syntax of formal languages using EBNF.
- Given a state diagram, the student will be able to implement a simple lexical analyzer.
- Given the EBNF description of a grammar, the student will be able to construct a parse tree for a sentence generated by the grammar.
- Given the EBNF description of a grammar, the student will be able to implement a simple recursive descent parser.
- The student will be familiar with shift-reduce parsing algorithms.
- The student will be familiar with the various kinds of bindings that occur during language design and implementation, program compilation, program load time, and program run time.
- The student will understand how data types are implemented in various programming languages.
- The student will understand how subroutine linkage and activation is implemented.

- In order to exhibit a basic understanding of functional programming languages, the student will be able to write simple Scheme programs consisting of multiple function definitions.
- In order to exhibit a basic knowledge of logic programming languages, the student will be able to write a simple Prolog program.

Assessment Instruments:

Programming assignments (60 %)
Exams (30 %)
Final Exam (10 %)

There will be three examinations. All exams will be comprehensive in nature.

Attendance Policy:

Attendance is expected as well as required for each lecture. A penalty of 3% will be deducted from the final course grade for each missed lecture in which attendance was recorded (instructors prerogative on a lecture by lecture basis).

Course Grading:

A: [90% - 100%]
B: [80% - 90%)
C: [70% - 80%)
D: [60% - 70%)
F: below 60%)

CS body of knowledge core topics (from CC2001):

- PL1: Overview of programming languages
- PL2: Virtual machines
- PL3: Introduction to language translation
- PL4: Declarations and types
- PL5: Abstraction mechanisms
- PL6: Object-oriented programming
- PL7: Functional programming
- PL--: Logic programming
- PL8: Language translation systems
- PL9: Type systems
- PL10: Programming language semantics
- PL11: Programming language design