CS430 Programming Languages
Exam 1 Study Guide

This exam will include matching, short answer, true/false, and/or code evaluation type questions. No extensive coding problems will be included. Exam questions will be similar to lab and class activity questions and problems.

The exam will cover the following learning objectives:

- Evolution of Programming Languages (Chapter 2)
    - Describe the general evolution of programming languages, and the major features of successive generations of languages
    - Explain how imperative languages reflect the Von Neumann architecture
    - Explain the significant features introduced by major languages such as Fortran, Algol, Cobol, C, and Java
    - Describe the major programming paradigms (imperative, functional, declarative) and their primary languages
    - Define important terms and concepts

- Syntax and Semantics (Chapter 3)
    - Define syntax, static semantics, and dynamic semantics
    - Define non-terminal, terminal, derivation, and parse tree
    - Interpret and write BNF descriptions of languages
    - Create derivations and parse trees for sentences of a given grammar
    - Create left-most and right-most derivations
    - Define ambiguity in a grammar, and modify a grammar to remove ambiguity
    - Define associativity in a grammar, and change associativity within a grammar
    - Define precedence in a grammar, and change precedence within a grammar

- Chapter 4: Parsing (Chapter 4)
    - Define syntax analysis, lexical analysis and parsing, including the purpose and product of each
    - Define token and lexeme, and relate them to lexical analysis

- Scripting Languages
    - Explain the characteristics of scripting languages that differentiate them from other general-purpose languages
    - Name several scripting languages
    - Explain the properties of the Perl language
    - Interpret Perl expressions

- Functional Languages (Chapter 15)
    - Describe the formal basis for functional languages
    - Explain the differences between functional and imperative languages related to design objectives, computational model, and implementation
    - Evaluate basic Dr.Racket expressions