Midterm Exam Study Guide

Lecture #1
1. Programming Domains
   a. Scientific Applications
   b. Business Applications
   c. Artificial Intelligence
   d. Web Software
2. The three main qualities of programming languages:
   a. <u>Readability</u> – includes simplicity, orthogonality, control structures, data types, syntax
   b. <u>Writability</u> – how does it relate to abstraction
   c. <u>Reliability</u> – type checking, exception handling, aliasing
3. Influences on programming language design:
   a. Computer architecture
   b. Programming methodologies
4. Four programming paradigms:
   a. Imperative (or procedural)
   b. Functional
   c. Declarative
   d. Object-Oriented
5. Translation issues: Compiling vs. interpreting vs hybrid

Lecture #2
1. The evolution of programming languages from machine language to Assembler to FORTRAN (I to II to IV to 77 to 95), COBOL, ALGOL and LISP to BASIC, PL/I and Pascal to C to C++, Java and C#.
2. Were PL/I, Ada successes or not? Why didn't Modula-2 catch on?
3. Why did
4. Why did some languages succeed (like FORTRAN and COBOL) and others fail (like APL)?
5. How did scripting languages (e.g., Perl, Python, PHP and Ruby) catch on?

Lecture #3
1. What separates lexical, Syntactic and semantic errors?
2. BNF
3. EBNF
4. Lexemes and Tokens
5. Parsing – Parse trees vs abstract syntax trees

Lecture #4
1. <u>Translators</u> – compilers, interpreters, and linkers
2. <u>Compilers</u> – top-down vs bottom-up parsers, scanners, parsers, semantic analysis, optimization, intermediate vs final code generation

3. Top-down parsing

Lecture #5
1. Why should it matter if names are case sensitive?
2. Why should it matter if special words are reserved words or keywords?
3. Does the length of a name matter?
4. Why do some languages include special characters in variable names?
5. What attributes do names typically have?
6. What are binding and binding time?
7. Static vs dynamic binding
8. Explicit vs Implicit declarations?
9. Dynamic vs static type binding
10. Static vs stack dynamic vs heap dynamic variables
11. Global vs local variables

Writing simple programs in C, C++, FORTRAN and BASIC