Objectives
To convey an understanding of the structure, use and formal description of a variety of Programming Languages. This includes Imperative, Object Oriented Functional, Logical, Multi-programming and perhaps other languages. Examples are drawn from C++, Scheme, Prolog, and others languages. The significance of language facilities such as Typing, Scoping, Recursion, Parameter Passing, etc. are covered. The material in this course is a necessary background for study of Compiler Construction.

Expected Work and Effect On Grade
HW ASSIGNMENTS ~1 per week From the book and elsewhere. (Good For Test Prep)
Grade Effect: Used in borderline cases

PROGRAMMING ASSIGNMENTS: 3 or more:
There will be 3 Programming Assignments+perhaps 1 for extra credit.
Grade Effect: 30%

MIDTERM EXAM:
Grade Effect: 30%

FINAL EXAM:
Grade Effect: 40%

NOTE: THE OVERALL GRADE IS BASED ON THE PERCENTAGES ABOVE, BUT YOU MUST ALSO DO PASSING WORK ON BOTH THE PROGRAM ASSIGNMENTS AND THE EXAMS TO PASS THE COURSE.
Lecture Notes will be provided for each lecture - generally provided before the lectures.

The text is comprehensive, covering the course material and more. For each lecture the relevant sections of the book which cover similar material will be indicated. On exams you are responsible for the intersection of the relevant lecture and book material. (This will usually be the material in the Notes)

OUTLINE

There will be detailed NOTES on-line for each Lecture

COURSE CONTENTS BY REFERENCE TO BOOK

Chapter 1 Language Design Issues

Chapter 2
  2.23 Bindings

Chapter 3 Translation Issues
  3.3
  3.3.1 BNF Grammar
  3.3.2 FSA and Regular Expressions
  3.3.3 Formal Properties-Chomsky Hierarchy

Chapter 4 Language Semantics
  4.1.1 Formal Properties Of Grammars
    Chomsky Hierarchy
  4.1.1 Only Ambiguity (Other won’t Hurt)
  4.2 Semantics
    4.2.1 Attribute Grammars

Chapter 4 Elementary Data Types

To Be Continued.