Catalog Description  Formal description of programming languages and techniques used in their compilation. Study of syntax, semantics, ambiguities, procedures replication, iteration, and recursion in these languages. Syntactic decomposition and the theory of compilers that are syntax-directed or recursively controlled.

Course Goals  1. Understanding the principles of language designs.
              2. Understanding of the top-down recursive descent compiling.
              3. Ability to implement minimal programming language processing system.


Prerequisites by Topic.  Working knowledge of a high level programming language; programming experience; some understanding of the underlying theory (formal languages) and computer architecture.

Major Topics Covered in the Course:
[1-2] History of programming languages; assemblers; macro-preprocessors; pre-theory language (FORTRAN); ALGOL; PL/I; Pascal/Modula; C; ADA.
[3-4] Compilers and Interpreters; Anatomy of a compiler; Passes and Phases; Overview of a sample compiler.
[8-9] Scanner.
[10-11] Context-free grammars; top-down parsing; bringing grammars to LL(1)- suitable form.
[12-13] The expression grammar; an expression evaluator; converting expressions to the postfix form. [14-15] The target hardware and code-generation overview (P-code machine); a simple model of converting P-code to a register code. Detailed analysis of common statements: [ 16] Simple statements (repeat, while, if).
[17] goto
[18] case
[22] type trees.
[23] Data-type conversion.
[24] Separate compilation and units.

Homework  Partial problems assignment related to the project will be assigned regularly. The grade will be based on the total.

Grading  The grading is based entirely on the success in the implementation of a small Pascal language compiler (100%). This is a large scale project; the students are expected to work on it through the semester. The Attendance may be taken.

Office Hours  My hours this term for CSc 420 will be 5:00 – 6:00 on Mondays.