CS 4090: LANGUAGE DESIGN AND IMPLEMENTATION

Course Description

Fundamental concepts of languages. Processors, data, operations, sequence control, data control, storage management, syntax, translation. Prerequisite: CS 2170 and CS 3350.

Course Syllabus

- **Introductory Concepts**
  - The Nature of Systems Software
  - Types of Language Translation Systems
  - Language Specification
  - Backus-Naur Form
  - Summary of Language Design Concepts

- **Compilers**
  - Token Representation (Internal Compiler Code)
  - Lexical Scan
  - Number Conversion
  - Name Declaration
  - Code Generation
  - Types of Compilers
  - Table Compression Algorithms

- **Implementation of Specific Language Features**
  - Arithmetic Expressions
  - Procedure calls
  - Parameter passing: by value, by reference, by name, by value/result
  - Recursion and Reentrancy
  - Multi-dimensional arrays (including subscript mapping functions)
  - Control structures (if/then/else, case, while, etc.)

- **Error Analysis and Debugging Aids**
  - Error Detection
  - Error Repair
  - Error Correction strategies (e.g. spelling correction)
  - Implementation strategies for symbolic debuggers
  - Code trace and variable trace alternatives

- **Symbol Table Organization**
  - Hashing, Linear, Binary, Linked Lists
  - Considerations for block-structured languages and various scope rule conventions

- **Role of Linker**
  - Linker responsibilities
  - Special requirements for object-oriented languages (such as name mangling)

- **Code Optimization**
  - Introduction to data flow analysis
  - Introduction to program flow analysis
  - Specific optimization algorithms (both machine-independent and machine-specific)

- **Differences between Compilers and Interpreters**

Course Project

Implementing a compiler for a small programming language.