CSCI 5341 - COMPILER PRINCIPLES AND TECHNIQUES

CREDIT HOURS: 3
PREREQUISITES: CSCI 3342; CSCI 4341 or 5340
GRADE REMINDER: Must have a grade of C or better in each prerequisite course.

CATALOG DESCRIPTION

Language theory, grammars and recognizers, methods for lexical analysis, top-down and bottom-up parsing, code generation, run-time structures, optimization, error handling.

PURPOSE OF COURSE

To present fundamental concepts of languages and the formal mechanisms for describing and translating languages, to become familiar with compiler-writing tools, and to study all phases of compiling with primary emphasis on parsing and generating code for high-level programming languages.

EDUCATIONAL OBJECTIVES

Upon successful completion of the course, students should be able to:

1. Describe elements of formal language theory that are directly pertinent to lexical analysis and syntax analysis.
2. Describe top-down and bottom-up parsing methods fundamental to performing lexical analysis and syntax analysis on programs written in typical high-level programming languages.
3. Design a compiler to generate object code corresponding to a high-level programming language source program.
4. Describe the importance of the use of formal grammar techniques in the specification of a variety of process control applications.

COURSE CALENDAR

This course meets for a minimum of 37.5 lecture contact hours during the semester. Students have significant assignments based on readings from the primary literature, participate in classroom discussions regarding current research topics, complete periodic homework and laboratory/programming assignments, and periodic exams in addition to the final exam. Students are expected to prepare for any class assignments or quizzes over the material covered in class or in the reading material. Successful completion of these activities requires at a minimum six additional hours of outside of classroom work each week.

CONTENT Hours

Survey of Programming Language Features and Overview of Compiling 2

Formal Language Notation and Classification of Languages.....5

Grammars.....5
- Regular grammars and lexical analysis
- Context free grammars and syntactic analysis
- Normal forms

Recognizers 5
- Finite state automata and finite state diagrams
- Pushdown automata

Parsing Techniques 15
- Top-down recursive descent parsing
- Predictive parsing
- Bottom-up parsing
- Shift/reduce methods
- Parse table generators
- Use of compiler development tools

Code Generation, Syntax-Directed Translation 5

Run-Time Storage Management 3

Error Handling: Detection and Recovery 2

Code Optimization: Loop Optimization, Data Flow Analysis 3

TOTAL 45

REFERENCES

Aho, Sethi, and Ullman, Compilers: Principles, Techniques, and Tools, Addison-Wesley, 1986.

Barrett, Bates, Gustafson, and Couch, Compilers Construction: Theory and Practice, Science Research Associates, 1986.

Harrison, M.A., Introduction to Formal Language Theory, Addison-Wesley, 1978.

Holub, A.I., Compiler Design in C. Prentice Hall, Englewood Cliffs, N.J., 1990.