

Syllabus
Department of Mathematics and Computer Science
CSC 460: Language Translation

A. Protocol

Course Name: Language Translation
Course Number: CSC 460
Credits: 3
Prerequisites: Co-Requisite CSC 475

Upon entering this course the student will be able to:

- a) Discuss and illustrate the definition of a data structure
- b) Discuss the concepts of pointers and linked lists
- c) Discuss, illustrate and write programs that use stacks, queues and trees
- d) Discuss the applications of specific data structures
- e) Discuss the management of computer memory using elementary abstract data structures
- f) Write programs which exemplify good programming style, efficiency of code and proper documentation
- g) Write large programs in modular fashion
- h) Allocate memory for integers of various sizes and perform operations on those memory locations
- i) Discuss and illustrate the allocation and manipulation of arrays using an assembly language
- j) Write and discuss various forms of indirect addressing and two-levels of indirect addressing modes
- k) Allocate characters locations and manipulate character data
- l) Discuss the concepts of subprograms in an assembly language, including parameter passing and recursion
- m) Allocate floating-point memory locations and perform floating-point operations
- n) Perform a binary search in a computer program.

Date of Revision: Spring 2010

B. Objectives of the Course

Upon completion of this course the student will be able to:

- a) Identify the phases of a compiler.
- b) Design a data structure to store tokens.
- c) Discuss the creation of symbol tables.
- d) Discuss error analysis of computer programs.
- e) Distinguish between lexical and syntactic errors.
- f) Write transition diagrams recognizing tokens.
- g) Specify tokens using regular expressions.
- h) Design a Lexical Analyzer using transition diagrams.
- i) Use syntax definition to analyze programs and statements.
- j) Use translation of postfix notation to analyze perform code generation of assignment statements.
- k) Discuss top down, recursive descent, and predictive parsing.
- l) Use predictive parsing to perform syntax analysis of control statements.

- m) Discuss the theory of grammars.
- n) Discuss syntax directed translation of control statements.
- o) Write programs performing Lexical Analysis, Syntactic Analysis, and Code Generation of high level programming languages.

C. Catalog Description

This course studies the design and construction of compilers. Lexical analysis, syntactic analysis, and code generation are investigated in detail. Language design, interpreters, semantic analysis, intermediate code generation, and code optimization are also considered. Prerequisite: CSC 323 Assembly Language with C- or better. Three credits.

D. Outline of the Course

- a) Overview of compiling
 - i) The phases of a compiler
 - ii) The lexical analyzer
 - iii) The syntactic analyzer
 - iv) The code generator
 - v) Error analysis
 - vi) Example class language to be compiled
- b) Lexical analysis
 - i) The role of the lexical analyzer
 - (1) Generation of a stream of tokens
 - (2) Creation of the symbol table
 - (3) Lexical error analysis
 - ii) Specification of tokens using regular expressions
 - iii) Recognition of tokens using transition diagrams or finite automata
- c) Syntax analysis
 - i) Syntax definition through grammars and parsing trees
 - ii) Syntax analysis of assignment statements
 - (1) Types of errors
 - (2) Postfix notation
 - (3) The jump table algorithm
 - iii) Syntax analysis of control statements
 - (1) Top down parsing
 - (2) Recursive descent parsing
 - (3) Predictive parsing
 - (4) Other parsing techniques
 - iv) Introduction to the theory of grammars
- d) Code generation
 - i) Syntax-directed translation
 - ii) Translation schemes and applications to assignment statements
 - iii) Translation of control statements
- e) Other possible topics
 - i) Interpreters
 - ii) Semantic analysis
 - iii) Intermediate code generation
 - iv) Code optimization
 - v) Compiler generators

E. Teaching Methodology

This course will be taught using the lecture/discussion method with demonstrations. Students will be required to design and write a simple compiler in C. At the discretion of the instructor, group projects may be given. Class lectures will stress methods for designing compilers

F. Assessment Activities

The final grade will be determined as a percentage from the following evaluation methods with varying weights at the discretion of the instructor:

- a) Written examinations
- b) Quizzes
- c) Assignments
- d) Programs
- e) Attendance
- f) Performance

G. Accommodations for Students with Disabilities

Students with disabilities:

- Reserve the right to decide when to self-identify.
- Must register with the Office for Students with Disabilities (OSD) to receive services.
- Will provide the appropriate notice from OSD for accommodations which specifically involve the faculty.

Students with disabilities receive services from the Office for Students with Disabilities (OSD). The OSD is located in the Azorsky Building, Room 105. The phone number is (724)938-5781. Requests for accommodations should be directed to this office and require the student to submit a completed Accommodation Request Form. Approved accommodations will be recorded on the Accommodation Approval Notice and provided to the student.