COMPILERS AND ASSEMBLERS

School of Health Sciences, Technology and Trades

Course Number: CST8152	**Contribution to Program:** Core	**Educator(s):**
Applicable Program(s): Computer Science Technology Computer Engineering Technology	**AAL:** 05 05	**Approval Date:** Winter 1997 **Approved By:** Title: David Fisher, Chairperson Computer Studies/Math Dept.
Course Hours: Delivered: Normative:	**Prerequisites:** CST8130, CST8134 **Corequisites:** None	**Approved for Academic Year:** 1996 - 1997

COURSE DESCRIPTION

This medium level course is essentially an applied course. The C language is used as a tool to create an interpreter/compiler. No new C constructs or C language elements are taught; rather, the language is used to develop a rather complex programming application.

RELATIONSHIP TO PROGRAM LEARNING OUTCOMES	
This is a vocational course that supports the following vocational program standards:	**This course contributes to your program by helping you to achieve the following provincial generic skills standards:**

COURSE CURRICULUM

I. Course Learning Requirements/Embedded Knowledge and Skills

Course Learning Requirements	Knowledge and Skills
When you have earned credit for this course you will have demonstrated an ability to:	
Understand the process of compiling and the parts of a compiler.	Program debugging using the Borland debugger. Use of Borland C Projects to compile and link multiple files containing various functions. Application of the C language in a complex program.
Understand the process of and tools for lexical analysis.	
Understand the tools for definition of programming language and grammars.	
Identify the different types of grammars and use a variety of parsing techniques.	
Describe the different aspects of code generation and the commonly used code optimization techniques.	
Describe the nature of a compiled and assembled program executable.	

II. Learning Resources

The course consists of three hours of lectures and one hour of lab per week.

Lab assignments will be done on the PC's in a lab equipped with Borland C. Assignments will reinforce the material covered in the lectures. Some assignments will reinforce the material covered in the lectures. Some assignments will usually have to be completed outside lab hours. Students will have access to the lab after hours for this purpose.

Textbook:

Aho, et al; Compilers - Principles, Techniques & Tools; Addison-Wesley

References:

Charles N. Fischer, R. J. LeBlanc; Crafting a Compiler with C; Addison-Wesley

Holub; Compiler Design in C; Prentice-Hall

Purchase of the textbook is **strongly** recommended as much of the lecture material closely follows and is supplemented by the textbook.

III. Teaching/Learning Methods

During this course you are likely to experience:

Lecture sessions will present the theoretical material of the course, aided by use of the overhead projector and brief lecture notes. Students will be expected to read and understand applicable material in the textbook. Students are encouraged to ask questions during lectures and to consult with the professor on topics which they do not clearly understand.

Laboratory sessions will provide opportunities for students to supplement the lecture material with a series of related assignments. Students will design, code and test components of a simple compiler. Students should seek advice and help from the professor in the laboratory.

IV. Learning Activities and Assessment

Samples of learning activities include:

Lectures will include analysis of compilers and language translation techniques. This will include lexical analysis, parsing, context-free grammars, code generation and optimizing techniques. Students are expected to observe carefully and to ask for clarifications or further examples.

Topics will include:

1. The compilation process (2 weeks)

   Relationship between languages and machines, aspects of the compilation process, assembly and linking, parts of a compiler.

2. Lexical analysis and language definition (4 weeks)

   Recognition of symbols, output from the lexical analyzer, state transition diagrams, finite state automata. Syntax and semantics, grammars, formal definition of programming languages, parse trees, the parsing problem.

3. Context-free grammars and syntax analysis (4 weeks)

   Context-free grammars, recursive descent parsing, bottom-up parsing.

4. Code generation (2 weeks)

   Production of quadruples, symbol table manipulation, number of passes, intermediate languages, target language.

5. Other aspects of compiling, assemblers (3 weeks)

   The run-time stack, object code optimization, types of error, error recovery and diagnostics. Assemblers and related programs, two-pass assembler, one-pass assembler.

Laboratories will provide opportunities for hands-on use of the computer to write, test and debug computer programs, with the professor in attendance and on call for assistance. Laboratories will also be used for individual demonstration and evaluation of completed work.

Students are expected to work on their own and to ask for assistance from the professor when necessary. Students may be required to show completed pre-lab portions of assignments before being admitted to the laboratory.

V. Evaluation/Earning Credit

The following will provide evidence of your learning achievement:

Assessment of student learning will be done by means of class quizzes, mid term tests, lab assignments, and a written final examination.

Lab attendance is mandatory, and absence from three or more laboratory sessions without the consent of the instructor will result in a final grade of F.

All lab assignments and the research project must be successfully completed in order to receive credit. Late assignments will be penalized: late less than 1 week - 20%, late 1 week or more - 100%.

The factors in the final grade are:

1.	Lab assignments	20%	
2.	Mid term tests		40%
3.	Final exam	40%	

The final grade will be calculated as the total of the above factors. In addition, the student must achieve at least 40 out of 80 in the total of the mid terms and final exam to receive a passing grade.

VI. Prior Learning Assessment

Evidence of learning achievement for PLA candidates will include:
Portfolio of related work completed by the student, completion of a challenge test with a breadth of coverage and level of difficulty equivalent to the final examination in the course, with a grade of at least C and at the assessor's discretion, successful completion of a special assignment relating to the course content.

RELATED INFORMATION

If you are a student with a disability please identify your needs to the professor and/or the Centre for Students with Disabilities (CSD) so that support services can be arranged for you. You can do this by making an appointment at the CSD, Room C142, Ext. 7683 or arranging a personal interview with the professor to discuss your needs.

Students, it is your responsibility to retain course outlines for possible future use to support applications for transfer of credit to other educational institutions.