COMPILERS

http://www.algonquinc.on.ca/cst/8152/98w/

School of Health Sciences, Technology and Trades

Course Number: CST8152	**Contribution to Program:** Core	**Educator(s):** Ian D. Allen alleni@algonquinc.on.ca
Applicable Program(s): Computer Science Technology Computer Engineering Technology	**AAL:** 05 05	**Approval Date:** 1998 Winter Semester **Approved By:** Shawn McBride, Chairperson, Computer Studies/Math Department mcbrids@algonquinc.on.ca
Course Hours: Delivered: 64 Normative: 64	**Prerequisites:** CST8130, CST8134 **Corequisites:** None	**Approved for Academic Year:** 1998

COURSE DESCRIPTION

This is a mid-level applied course with a heavy emphasis on program structure and programming. The theory of compilers provides the foundation to develop a moderately complex application. The C programming language is used as a tool to create a series of increasingly powerful text processors / interpreters. Students will learn how to structure and test a mid-size software project (1000-2000 lines). Students will use their knowledge of data structures such as stacks, tables, strings, and dynamic memory allocation to implement their projects. Each assignment builds on the previous, so the timely and thoughtful completion of each and every assignment is essential.

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:	**You will exercise the following skills:**
Describe the nature of a compiled, assembled, and linked program executable, including static, global, and local data, parameter passing details, and the C run-time environment.	Program debugging. Use of dynamic memory. Text string handling and parsing. Coordinating the compilation and linking of multiple files containing local, static, and global data structures and functions. Application of the C programming language to a complex program. Writing understandable large programs. Building code for modification and re-use. Modularity. Defensive programming.
Understand the process of compiling and the parts of a compiler.	
Understand the process of and tools for lexical analysis, parsing, error handling, and interpretation.	
Understand basic grammars and regular expressions. Structure a mid-size programming project.	

II. Learning Resources

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

Assignments derive from the material covered in the lectures. Due to their complexity, most assignments require *substantial* work outside assigned laboratory hours. Students will have access to the lab after hours for this purpose.

Lab assignments may be developed using any ANSI C programming environment; but, they must compile and run in the Algonquin laboratory environment. Laboratories are equipped with Borland and Turbo C compilers running under Windows 95.

Assignments written in languages other than ANSI C are not accepted; assignments that only work at home or elsewhere, but do not work in the labs, are not accepted.

Textbook:

Aho, et al; *Compilers - Principles, Techniques & Tools*; Addison-Wesley, 1986

References:

Internet Class Notes: **`http://www.algonquinc.on.ca/cst/8152/98w/`**

III. Teaching/Learning Methods

Lecture sessions, course notes, and assigned readings will present the theoretical material of the course, aided by the use of support media such as overhead transparencies or computer demonstrations. Students are expected to read assigned material and be prepared to answer oral or written questions in following lectures and laboratories. Laboratory sessions will require

students to apply the readings and lecture material to a series of assignments. The assignments build upon each other and become increasingly complex as the course progresses. Students are encouraged to ask questions during lectures and laboratories.

IV. Learning Activities and Assessment

Lectures and readings will touch on the following topics. Students are expected to observe carefully and to ask for clarifications or further examples:

1. The compilation process and run-time environment

   Interpreters vs. compilers. Relationship between languages and machines. The compilation process, parts of a compiler. Linking. Static and global data. The run-time stack, parameter passing, memory management.

2. Lexical analysis and language definition

   Recognition of symbols, output from the lexical analyzer, state transition diagrams, finite state automata, regular expressions. Syntax and semantics, grammars, parse trees, the parsing problem.

3. Syntax analysis, parsing and semantics

   Recursive descent parsing, error recovery, symbol tables, type checking. Coding semantic routines.

4. Other aspects of compiling (as time permits)

   Code generation, intermediate languages. Object code optimization, parser generators.

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.

Completed programming assignments will be collected in machine-readable form for analysis and assessment. Students may be required to submit the assignments via electronic mail, floppy disks, or over the Internet. Printed copies may also be required.

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, lab demonstrations, and a written final examination.

Assignments will be penalized for lateness: late less than 1 week, -20%; late 1 week or more, -100%. All laboratory assignments *must be successfully completed* to receive credit for the

course, even if the assignments are late.

The factors in the final grade are:

1.	Lab assignments	40%
2.	Mid term tests	30%
3.	Final exam	30%

Later assignments are worth more than early assignments, since each assignment builds on the previous. The final grade will be calculated as the total of the above factors. In addition, the student must achieve at least 30 out of 60 in the total of the mid terms and final exam to receive a passing grade.

Algonquin College has severe penalties for people who claim authorship of material they did not write:

Plagiarism is defined as:

> *Material submitted by the student, claimed as the student's own work, that is substantially the work of one or more other people. Plagiarised material may come from such places as other student disks, source listings, textbook examples, and course notes. If the material is copied without clear mention of its true origin, the material is plagiarised*.

The first occurrence of plagiarism will result in a mark of **zero** for the assignment. All the students involved will be required to **redo** the copied assignment. *An additional assignment* may be assigned to ensure that the students have mastered the material. No marks will be awarded for the additional assignment; but, *the assignment must be completed satisfactorily to receive credit in the course*. A second occurrence of plagiarism will result in an overall course grade of **F** for all concerned.

Students who knowingly allow their work to be copied (by lending diskettes or source listings) will receive the *same* sanctions as the plagiariser. Do not share your disks or listings with other students.

Share your ideas, not your source code.

VI. Prior Learning Assessment

Evidence of learning achievement for PLA candidates will include:

A 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 midterms and final examinations in the course, with a grade of at least C. At the assessor's discretion, successful completion of a special assignment (typically, a major programming 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.