Compilers

Information and Communications Technology

Course Number:	Co-Requisites:	Pre-Requisites:
CST8152	N/A	CST8234
Applicable Program(s):	**AAL:**	**Core/Elective:**
0006X01FWO - Computer Eng. Technology - Comp. Science	4	Core
0006X03FWO - Computer Eng. Technology - Comp. Science	4	Core
Prepared by:	Svillen Ranev, Professor	
Approved by:	Andrew Pridham, Academic Chair, ICT	
Approval Date:	Wednesday, June 24, 2015	
Approved for Academic Year:	2015-2016	
Normative Hours:	75.00	

Course Description

Introduction to the basic principles, techniques, and tools used to translate text expressed in one language to equivalent text expressed in another language. The concepts discussed and the programming concepts studied in previous courses are applied to develop and program the front-end of a simple compiler or interpreter using ANSI C as implementation language. The ideas and techniques discussed could be applied to general software design and to parsing of structured files, such as HTML, XML, register and configuration files.

Relationship to Vocational Learning Outcomes

This course contributes to your program by helping you achieve the following Vocational Learning Outcomes:

0006X01FWO - Computer Eng. Technology - Comp. Science

VLO 6	Analyze, build, test, implement, and maintain applications. (T,A)
VLO 8	Articulate, defend, and conform to workplace expectations found in technology environments. (T,A)
VLO 9	Contribute to the successful completion of the project applying the project management principles in use. (T,A)

0006X03FWO - Computer Eng. Technology - Comp. Science

VLO 6	Analyze, build, test, implement, and maintain applications. (T,A)
VLO 8	Articulate, defend, and conform to workplace expectations found in technology environments. (T,A)
VLO 9	Contribute to the successful completion of the project applying the project management principles in use. (T,A)

Relationship to Essential Employability Skills

This course contributes to your program by helping you achieve the following Essential Employability Skills:

EES 1	Communicate clearly, concisely and correctly in the written, spoken and visual form that fulfills

	the purpose and meets the needs of the audience. (A)
EES 2	Respond to written, spoken or visual messages in a manner that ensures effective communication. (A)
EES 7	Analyze, evaluate and apply relevant information from a variety of sources. (T,A)
EES 9	Interact with others in groups or teams in ways that contribute to effective working relationships and the achievement of goals. (A)
EES 10	Manage the use of time and other resources to complete projects. (A)
EES 11	Take responsibility for one's own actions, decisions and consequences. (A)

Course Learning Requirements/Embedded Knowledge and Skills

When you have earned credit for this course, you will have demonstrated the ability to:

1.) Describe the nature of a programming language and the process of converting a program written in a high-level programming language into executable code.

identify the theoretical and practical differences between existing programming paradigms and programming languages;

describe the function and the operation of editors, preprocessors, code generators, code optimizers, librarians, linkers, and other compiler supporting tools;

describe the function and the operation of the building blocks of Integrated Development Environments;

explain the how and when different compiler outputs and run-time environments are used. Detail the difference between compiler and interpreter.

2.) Explain the process of compiling a computer program and the parts of a compiler.

identify the main building blocks of a compiler and an interpreter;

explain the role of the lexical analyzer (scanner), syntax analyzer (parser), code optimizer, and code generator.

3.) Differentiate the process of and the tools for lexical analysis, parsing, error handling, and interpretation.

explain the main characteristics of a programming language and how it relates to a natural language;

identify and describe the different techniques and tools used for lexical, syntactical, and semantic analysis of a program written in specific language.

4.) Understand regular expressions and basic programming and scripting language grammars.

identify the role of the descriptive notations like regular expressions and grammars to describe a programming or scripting language;

use regular expressions to define the lexical part of a programming language;

use BNF grammar to define the syntactical part of a programming language;

implement a lexical analyzer (scanner) based on regular expressions;

use syntax-directed translation approach to implement a syntax analyzer (parser) based on a grammar description..

5.) Build and manage a mid-size programming project.

use the C programming language to write a complex program implementing the front end of a compiler;

use different compiler platforms to write effective, compact androbust multi-file programs;

coordinate properly the compilation and linking of multiple files containing local, static and global data structures and functions;

build code for modification and re-use applying modularity and defensive rogramming.

select and use appropriate tools and technology for building a project.

Learning Resources

Recommended:
1) Compilers – Principles, Techniques & Tools, 2nd ed., by Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Published by Pearson (Addison Wesley), ISBN-10: 0321486811

2) The C (ANSI C) Programming Language , 2nd ed., by Brian W. Kernighan, Dennis M. Ritchie, Published by Prentice Hall, ISBN 0-13-110362

This course is part of the BYOD (laptop) program initiative at Algonquin College. Students are required to have a functioning laptop at all lecture and lab classes. The specifications for the required laptop and additional information about the mobile program initiative can be found at: http://www.algonquincollege.com/byod/.

Software Required:
MS Visual Studio 2012/2013 or any ANSI C compliant compiler.

Additional Resources:
Blackboard Learn
Svillen Ranev's Home Page: http://elearning.algonquincollege.com/coursemat/ranevs

Learning Activities

- classroom lectures,
- software demonstrations,
- laboratory work,
- individual and team work
- practical assignments,
- research of course-related material.

The course consists of 3 hours of lectures and 2 hours of lab per week. It is anticipated that you will need to spend an additional 5 hours per week, on average, of your own time for assignments and study.

Classroom Lectures:

Lectures will present the theoretical material of the course. Students are expected to attend all of the lectures. Students are strongly encouraged to read the applicable material before coming to the lectures, and to be prepared to answer questions in lectures. Students are encouraged to ask questions during lectures and to consult with the professors on topics, which they do not clearly understand. Professors will inform students, at the beginning of the course, of suitable times for consultations.

Labs:

Attendance at the beginning of each laboratory period is compulsory, and absence from three or more laboratory sessions without the prior consent of the professor will result in a final grade of "F". Students are responsible for

keeping a record of the number of laboratory sessions they have missed. Professors will not inform students of an impending failure because of missed laboratory sessions.

The students will apply compiler theory to implementing the front end of a compiler for a case-study procedural programming language. The language is small and simple but includes many features which can be found in real programming, scripting, and descriptive languages. The students will be given the informal language specification. They are expected to write a grammar for the language and through a series of interrelated assignments to build the front end of the compiler. The assignments include implementation of some major compiler components like input buffer, scanner, symbol table, parser, and error recovery.

The ANSI C programming language will be used as a language of implementation. This will allow the student to master the language to great extent and to acquire expertise in the language implementation in different integrated development environments and compilers. Some features of the C language not covered in the previous courses will be presented and discussed in lab time.

Assignments build on earlier work, and are increasingly complex as the course progresses. Due to their complexity, most assignments require work outside assigned laboratory hours. Students may be allowed to work in recognized teams on some of the assignments. Assignment must be an authentic individual or a recognized team work. Late assignments will be penalized.

Laboratories will provide opportunities for hands-on use of the computer to write, compile, link, run, test and debug computer programs, with the professor in attendance. Laboratories may also be used for individual demonstration and evaluation of completed work or work in progress. Students are expected to work on their own and to ask for assistance from the professor when necessary.

Lab assignments may be developed using any operating system and **ANSI C** compliant programming environment and; but, they must compile and run in a specified test-bed environment, which includes a test-bed compiler and a test-bed run-time environment. Assignments written in languages other than **ANSI C** are not accepted.

Evaluation/Earning Credit

The following list provides evidence of this course's learning achievements and the outcomes they validate:

Midterm Exam(s) (30%)

Validates Outcomes: CLR 1, CLR 2, CLR 3, CLR 4, EES 2

Final Exam (30%)

Validates Outcomes: CLR 1, CLR 2, CLR 3, CLR 4, EES 2

Assignment(s) (40%)

Validates Outcomes: CLR 1, CLR 2, CLR 3, CLR 4, CLR 5, EES 1, EES 2, EES 7, EES 9, EES 10, EES 11

Prior Learning Assessment and Recognition

Students who wish to apply for prior learning assessment and recognition (PLAR) need to demonstrate competency at a post-secondary level in all of the course learning requirements outlined above. Evidence of learning achievement for PLAR candidates includes:

- Challenge Exam

- Project/Assignment

Grade Scheme

Final Grade	Mark Equivalent	Numeric Value	Final Grade	Mark Equivalent	Numeric Value
A+	90% - 100%	4.0	A	85% - 89%	3.8
A-	80% - 84%	3.6	B+	77% - 79%	3.3
B	73% - 76%	3.0	B-	70% - 72%	2.7
C+	67% - 69%	2.3	C	63% - 66%	2.0
C-	60% - 62%	1.7	D+	57% - 59%	1.4
D	53% - 56%	1.2	D-	50% - 52%	1.0
F	0% - 49%	0	FSP	0	0

Course Related Information

In order to pass the course, the student must have a grade of at least **50%** or "**D-**" on the midterm exam and final exam combined, and must have a grade of at least **50%** or "**D-**" on the programming project. (Students who have a failing grade on the combined midterm exam and the final exam will receive a grade of "**F**".)

Midterm exam and final exam combined will not be included in the final grade unless the student achieves at least a grade of **50%** or "**D-**" on the programming project.

The students' ability to successfully complete the programming project will directly correlate with their level of success on the midterm exam, on the final exam, and on the entire course.
Students will work on a programming project consisting of a series of four assignments which derive from the material covered in the lectures. Completed programming assignments will be collected in machine-readable form for analysis and assessment. Printed copies will also be required.

The programming project will consist of 4 programming assignments. All assignment must be successfully completed in order to obtain credit for the course. Each assignment builds on the previous, so the timely and thoughtful completion of each an every assignment is essential. Late assignments will be penalized. Each assignment will be evaluated and marked independently, but at the end the student must submit a working project in order to receive credit for the course. Working project is a project that compiles with the specified test-bed compiler, runs in the specified run-time environment, works according to the functional specifications, and passes all the tests specified in the individual assignments. In the case of a documented emergency the professor, in consultation with the Chair, will determine how the marks will be made up and/or final grade adjusted.
The Computer Studies Department requires that all course assignments (homework exercises, laboratory work, projects, etc) be submitted by students using a standard which could be specific to one or more courses. Professors will ensure, at the beginning of the term, that students are advised of the exact details of these course specific submission requirements. Professors will also post them online. Student submissions that do not meet the course published submission standards may not be marked, and may incur a penalty of up to 100% of the submission mark.
All students are required to write the final exam. There are no provisions for 'making up' a missed final exam. If, as a result of being off-track in your program or some unforeseen circumstance, you note that there is a scheduling conflict in your final exam schedule, it is your responsibility to alert your course professor no later than one week

before final exams start, to allow for any special arrangements

Department Related Information

STUDENT ACADEMIC RESPONSIBILITIES

Each student is responsible for:
• Knowing the due dates for marked out-of-class assignments.
• Attending all classes and knowing the dates of in-class marked assignments and exercises.
• Maintaining a folder of all work done in the course during the semester for validation claims in cases of disagreement with faculty.
• Keeping both paper and electronic copies of all assignments, marked and unmarked, in case papers are lost or go missing.
• Regularly checking both Blackboard announcements as well as one's Algonquin e-mail account for important messages from both professors and college administration.
• Participating in on-line and classroom exercises and activities as required.
• Retaining course outlines for possible future use to support applications for transfer of credit to other educational institutions.

Harassment/Discrimination/Violence will not be tolerated. Any form of harassment (sexual, racial, gender or disability-related), discrimination (direct or indirect), or violence, whether involving a professor and a student or amongst students, will not be tolerated on the college premises. Action taken will start with a formal warning and proceed to the full disciplinary actions as outlined in Algonquin College Policies - HR22 and SA07.
Harassment means one or a series of vexatious comment(s) (whether done verbally or through electronic means), or conduct related to one or more of the prohibited grounds that is known or ought reasonably to be known to be unwelcome/unwanted, offensive, intimidating, derogatory or hostile. This may include, but is not limited to: gestures, remarks, jokes, taunting, innuendo, display of offensive materials, offensive graffiti, threats, verbal or physical assault, stalking, slurs, shunning or exclusion related to the prohibited grounds.

For further information, a copy of the official policy statement can be obtained from the Student Association.

Violation of the Copyright Act
General – The Copyright Act makes it an offence to reproduce or distribute, in whatever format, any part of a publication without the prior written permission of the publisher. For complete details, see the Government of Canada website at http://laws.justice.gc.ca/en/C-42 . Make sure you give it due consideration, before deciding not to purchase a textbook or material required for your course.

Software Piracy - The Copyright Act has been updated to include software products. Be sure to carefully read the licensing agreement of any product you purchase or download, and understand the terms and conditions covering its use, installation and distribution (where applicable). Any infringement of licensing agreement makes you liable under the law.

Disruptive Behaviour is any conduct, or threatened conduct, that is disruptive to the learning process or that interferes with the well being of other members of the College community. It will not be tolerated. Members of the College community, both students and staff, have the right to learn and work in a secure and productive environment. The College will make every effort to protect that right. Incidents of disruptive behaviour must be reported in writing to the departmental Chair as quickly as possible. The Chair will hold a hearing to review available information and determine any sanctions that will be imposed. Disciplinary hearings can result in penalties ranging from a written warning to expulsion.

For further details, consult the Algonquin College Policies AA32, SA07 and IT01 in your Instaguide.

College Related Information

Email

Algonquin College provides all full-time students with an e-mail account. This is the address that will be used when the College, your professors, or your fellow students communicate important information about your program or course events. It is your responsibility to ensure that you know how to send and receive e-mail using your Algonquin account and to check it regularly.

Students with Disabilities

If you are a student with a disability, it is strongly recommended that you identify your needs to your professor and the Centre for Students with Disabilities (CSD) or Student Services, by the end of the first month of the semester in order that necessary accommodations or support services can be arranged for you.

Academic Integrity & Plagiarism

Adherence to acceptable standards of academic honesty is an important aspect of the learning process at Algonquin College. Academic work submitted by a student is evaluated on the assumption that the work presented by the student is his or her own, unless designated otherwise. For further details consult Algonquin College Policies AA18: Academic Dishonesty and Discipline and AA20: Plagiarism

Student Course Feedback

It is Algonquin College's policy to give students the opportunity share their course experience by completing a student course feedback survey for each course they take. For further details consult Algonquin College Policy AA25: Student Course Feedback

Use of Electronic Devices in Class

With the proliferation of small, personal electronic devices used for communications and data storage, Algonquin College believes there is a need to address their use during classes and examinations. During classes, the use of such devices is disruptive and disrespectful to others. During examinations, the use of such devices may facilitate cheating. For further details consult Algonquin College Policy AA32: Use of Electronic Devices in Class

Transfer of Credit

It is the student's responsibility to retain course outlines for possible future use to support applications for transfer of credit to other educational institutions.

Note: *It is the student's responsibility to refer to the Algonquin College Policies website for the most current information at http://www.algonquincollege.com/directives/*

Legend

Terms

- ALO: Aboriginal Learning Outcome
- Apprenticeship LO: Apprenticeship Learning Outcome
- CLR: Course Learning Requirement
- DPLO: Degree Program Learning Outcome
- EES: Essential Employability Skill
- EOP: Element of Performance
- GELO: General Education Learning Outcome
- LO: Learning Outcome
- PC: Program Competency
- PLA: Prior Learning Assessment
- PLAR: Prior Learning Assessment and Recognition

•VLO: Vocational Learning Outcome

Assessment Levels

•T: Taught

•A: Assessed

•CP: Culminating Performance