Lect
01

CST8152
Compilers

Algonquin College

Computer Engineering
Technology

CST8152
Compilers

Fall, 2023

Lect
01

Prof. Paulo Sousa

Algonquin College

Computer Engineering Technology

CST8152 Compilers

Fall, 2023

Lect 01

About the Course

Prof. Paulo Sousa

CST8152 — Compilers

Week 1: Presenting Compilers

- *Initial comments*
- *Before we start*
- *Course Overview*

L1	L2
A2W01	A2W09
A2W02	A2W10
A2W03	A2W11
A2W04	A2W12
A2W05	A2W13
A2W06	A2W14
A2W07	A2W15

A2W08

ALGONQUIN
COLLEGE

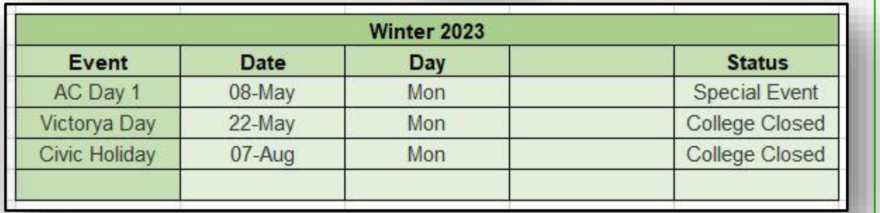
CST8152 — Compilers

Compilers – Week 1

Initial Comments

Welcome back to Campus

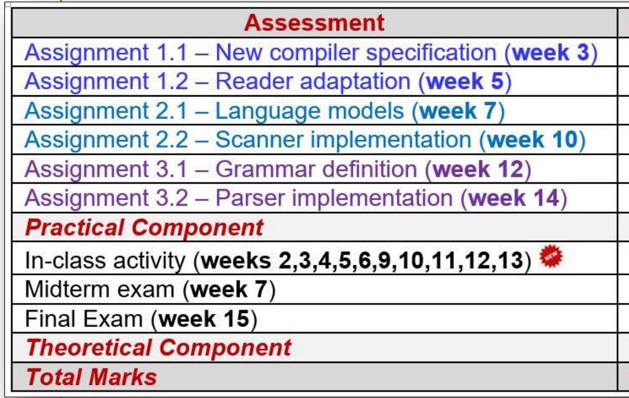
- AC gives you welcome to this new term!
  - *Time to learn and develop new skills!*



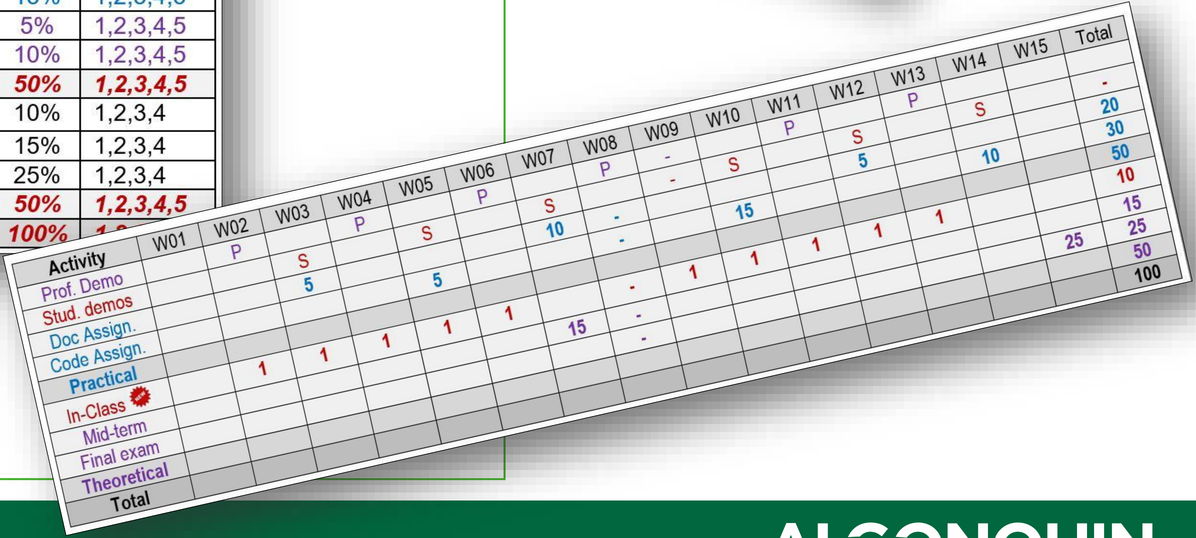
Winter 2023				
Event	**Date**	**Day**		**Status**
AC Day 1	08-May	Mon		Special Event
Victorya Day	22-May	Mon		College Closed
Civic Holiday	07-Aug	Mon		College Closed

Welcome

- Remember assessments / dates:



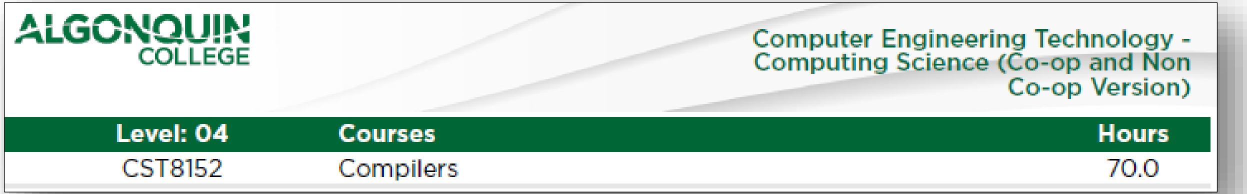
Assessment	Mark	CLRs
Assignment 1.1 – New compiler specification (**week 3**)	5%	1,2,3,4,5
Assignment 1.2 – Reader adaptation (**week 5**)	5%	1,2,3,4,5
Assignment 2.1 – Language models (**week 7**)	10%	1,2,3,4,5
Assignment 2.2 – Scanner implementation (**week 10**)	15%	1,2,3,4,5
Assignment 3.1 – Grammar definition (**week 12**)	5%	1,2,3,4,5
Assignment 3.2 – Parser implementation (**week 14**)	10%	1,2,3,4,5
Practical Component	***50%***	***1,2,3,4,5***
In-class activity (**weeks 2,3,4,5,6,9,10,11,12,13**)	10%	1,2,3,4
Midterm exam (**week 7**)	15%	1,2,3,4
Final Exam (**week 15**)	25%	1,2,3,4
Theoretical Component	***50%***	***1,2,3,4,5***
Total Marks	***100%***	***1,2***



Activity	W01	W02	W03	W04	W05	W06	W07	W08	W09	W10	W11	W12	W13	W14	W15	Total
Prof. Demo		P		P		P		P	-		P		P			
Stud. demos			S		S		S		-	S		S		S		-
Doc Assign.			5				10	-				5				20
Code Assign.					5			-		15				10		30
Practical																50
In-Class		1	1	1	1	1		-	1	1	1	1	1			10
Mid-term							15	-								15
Final exam								-							25	25
Theoretical																50
Total																100

We are here!

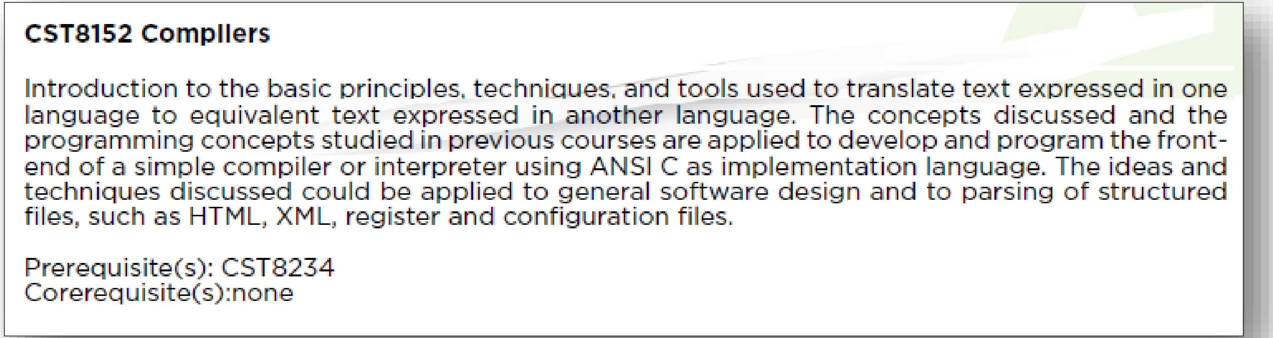
- **Level 4:**
  - 14 weeks;
  - 70.0 hours;
  - Model: (3/2/5)
- **Prerequisite**: C Language.
- **Titular Professor / Lab Professor:**
  - Paulo Sousa



ALGONQUIN COLLEGE

Computer Engineering Technology -
Computing Science (Co-op and Non
Co-op Version)

Level: 04	Courses	Hours
CST8152	Compilers	70.0



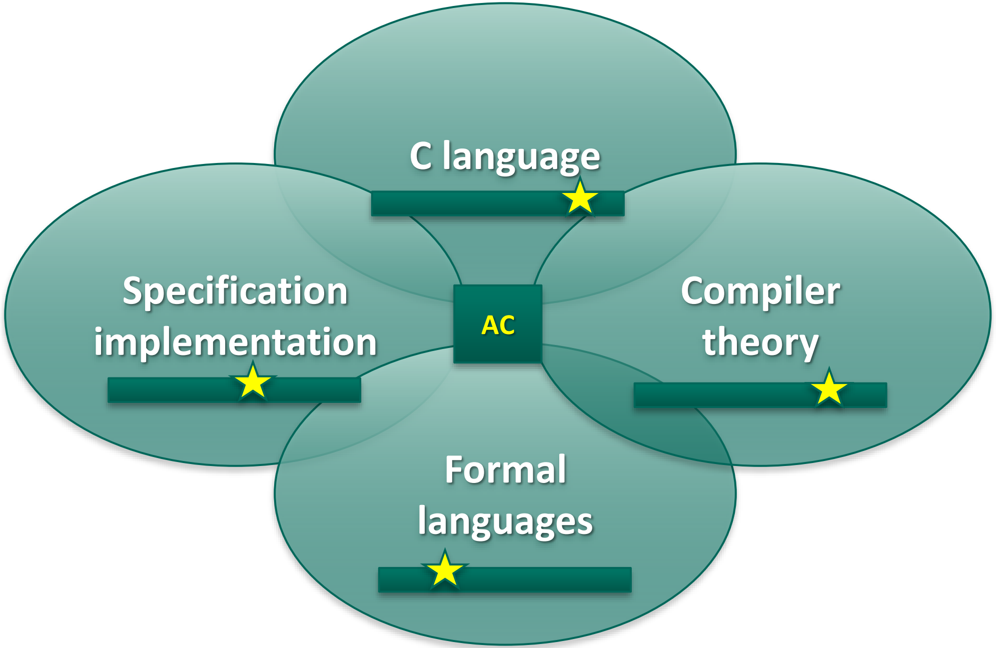
CST8152 Compilers

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.

Prerequisite(s): CST8234
Corerequisite(s):none

Source: Algonquin, 2019

The Zoo…



C language

Specification implementation

AC

Compiler theory

Formal languages

Compilers – Week 1

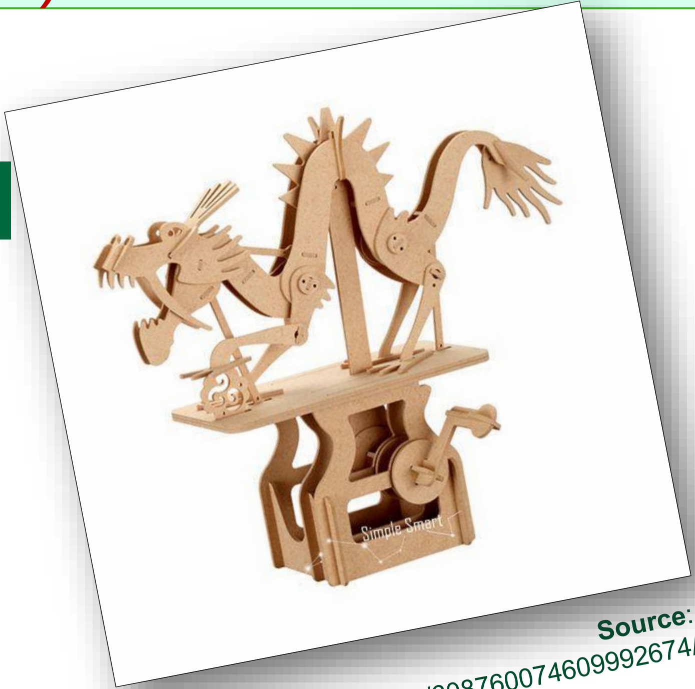
Before we start...

First Survey: Experien-C (Fall, 2023)

1. Please, access:

https://www.surveymonkey.com/r/DCDKSHB

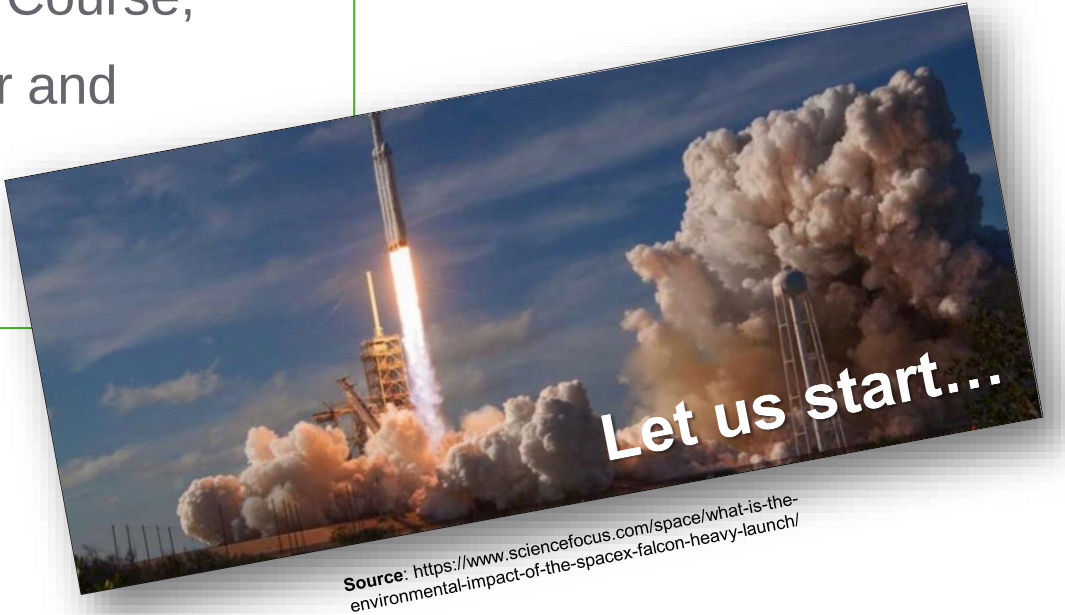
Source: https://medium.com/swlh/10-things-every-programmer-should-know-26ba37cfcaf4



Source: https://br.pinterest.com/pin/608760074609992674/

Weekly Outcomes

1. Getting Ready for this Course;
2. Knowing the Professor and Dynamics;
3. Preparing to Start…



Source: https://www.sciencefocus.com/space/what-is-the-environmental-impact-of-the-spacex-falcon-heavy-launch/

Good and Bad News

- Brief review…

Firstly, bad…

- Due their complexity, assignments really require work extra hours for development;
- Late assignments will be penalized following the rules from CSI / Standard Submission.;
- Other languages different from ANSI C will not be accepted and the MS Visual Studio 2019 is the default .

But there are good ones…

- The programming language to be implemented is simple;
- ANSI C gives you an expertise for the most different development environments;
- Features not covered in previous courses are presented and discussed in the Labs;
- Labs will provide hands-on opportunities to write and test the programs with professor assistance.

- Message:

Consider this Course as an opportunity to demonstrate your C-Language skills.

Compilers Lab Dynamics

- **Step-by-step:**

1. Each lab activity is related to a specific Assignment (or extra activity demanded by lecture professor);
2. The activities are progressive, which means that they are necessary to the next assignments.
3. During this time, to get bonus, you **need to present** the development of your activities that should satisfy some criteria about assignments.
   - These presentations are necessary to get full marks.
   - Doubts and suggestions can be discussed with the lab professor.

Source: https://www.algonquincollege.com/pembroke/program/computer-systems-technician/gallery/

Code of Conduct

- **Beyond the Code…**

1. No Harassment / Discrimination / Violence;
2. No infringement of Copyright Act;
3. No permission to Software Piracy;
4. No plagiarism and cheating risks.
5. Respect to Algonquin College Policies AA32, SA07 and IT01.

- **Important:**

No copies are allowed between the individuals / teams.

Source:https://www.cdc.gov/ethics/images/Ethics.png

Compilers – Week 1

Course Topics

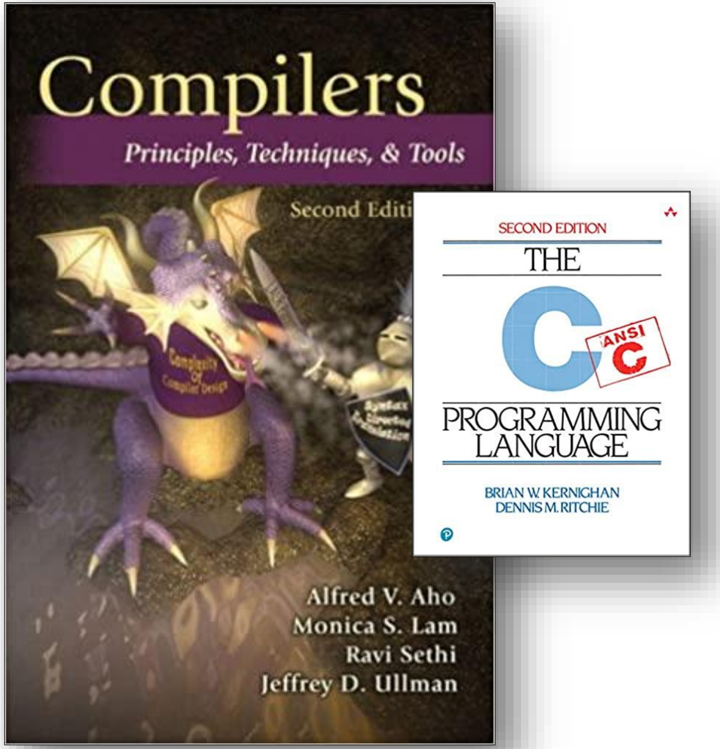
Course Topics

- Compilers
  - General View – Parts - Components
- Languages
  - Definition – Representations
- Front end-compiler (Analysis)
  - Lexical Analysis – Syntax - Semantic
- Practical aspects
  - C Language - Tools

Source: https://techcrunch.com/2016/05/10/please-dont-learn-to-code/

Course Textbook

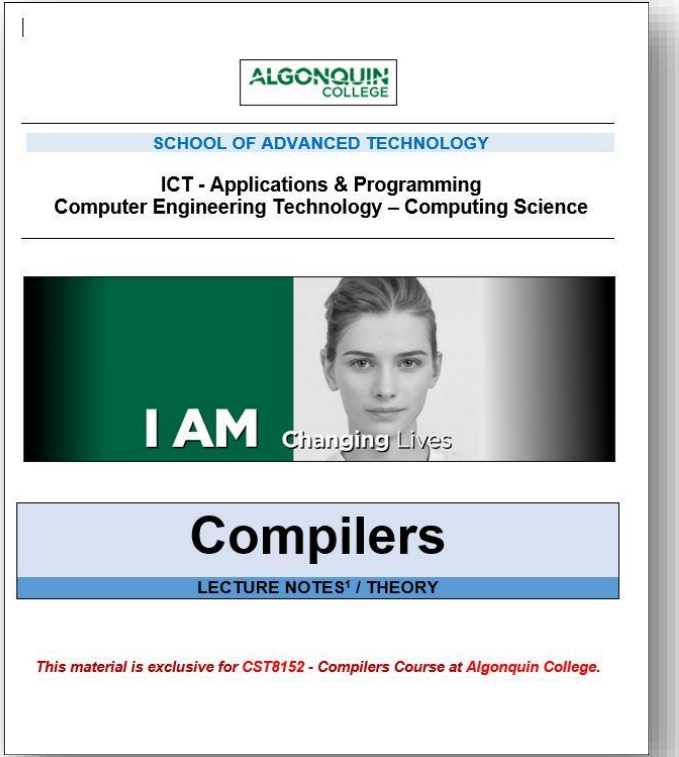
- Main Reference:
  - ***Compilers – Principles, Techniques & Tools***, 2nd ed., Pearson (Addison Wesley).
- Other references
  - **The C (ANSI C) Programming Language**, 2nd ed., by Brian W. Kernighan, Dennis M. Ritchie, Prentice Hall.



Course Lecture Notes

- Lectures will follow:
  - ***Compilers – Lecture Notes***, 2023 Fall version.

Lecture notes (originally created by prof. Svillen Ranev).



SCHOOL OF ADVANCED TECHNOLOGY

ICT - Applications & Programming
Computer Engineering Technology – Computing Science

I AM Changing Lives

Compilers

LECTURE NOTES[1] / THEORY

This material is exclusive for CST8152 - Compilers Course at Algonquin College.

Grades – Remembering…

Team activities - assignments

- **Axy** (5% + 5% + 10% + 15% + 5% + 10%)

Individual activities - theoretical

- Mid-term exam (20%) (5% from in-class activities)
- Final Exam (30%) (5% from in-class activities)

To pass:
[1] You need to achieve at least 50% of each component: 25 pts from Labs and 25 pts from Exams.
[2] In your final exam, you need to get at least half: 15 pts (from 30).

50%+50%

Source: https://www.sciencefocus.com/space/what-is-the-environmental-impact-of-the-spacex-falcon-heavy-launch/

Remember that you need to achieve at least 50% of each part in order to pass.

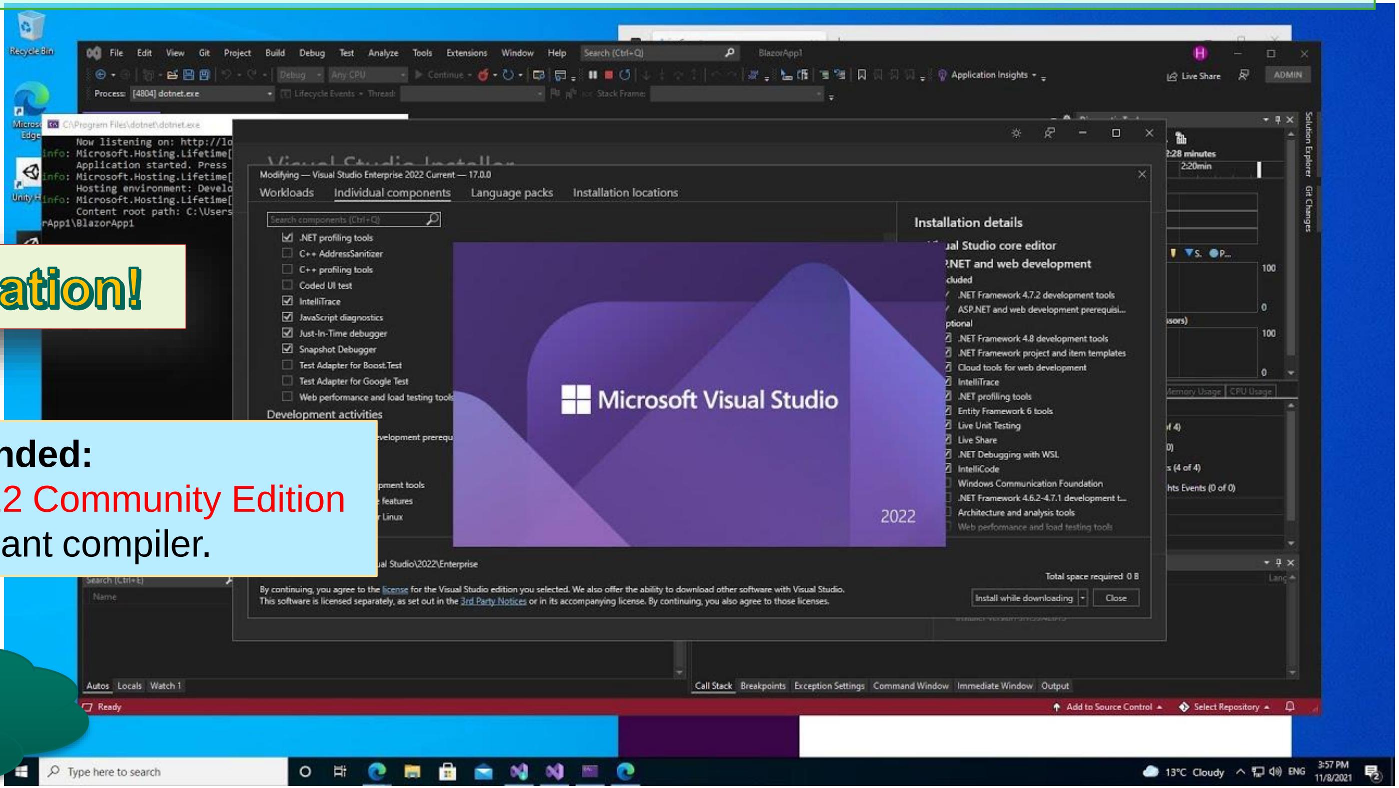
Work

- Lectures: 3 hours
- Weekly Labs – 2 hours
- Assignments
- Exams

The assignments are progressive. It means that you need to finish A(x) before going to A(x+1).

Source: https://medium.com/swlh/10-things-every-programmer-should-know-26ba37cfcaf4

Programming



```
Now listening on: http://lo
info: Microsoft.Hosting.Lifetime[
      Application started. Press
info: Microsoft.Hosting.Lifetime[
      Hosting environment: Develo
info: Microsoft.Hosting.Lifetime[
      Content root path: C:\Users
rApp1\BlazorApp1
```

Modifying — Visual Studio Enterprise 2022 Current — 17.0.0

- ☑ .NET profiling tools
- ☐ C++ AddressSanitizer
- ☐ C++ profiling tools
- ☐ Coded UI test
- ☑ IntelliTrace
- ☑ JavaScript diagnostics
- ☑ Just-In-Time debugger
- ☑ Snapshot Debugger
- ☐ Test Adapter for Boost.Test
- ☐ Test Adapter for Google Test
- ☐ Web performance and load testing tools

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 0 B

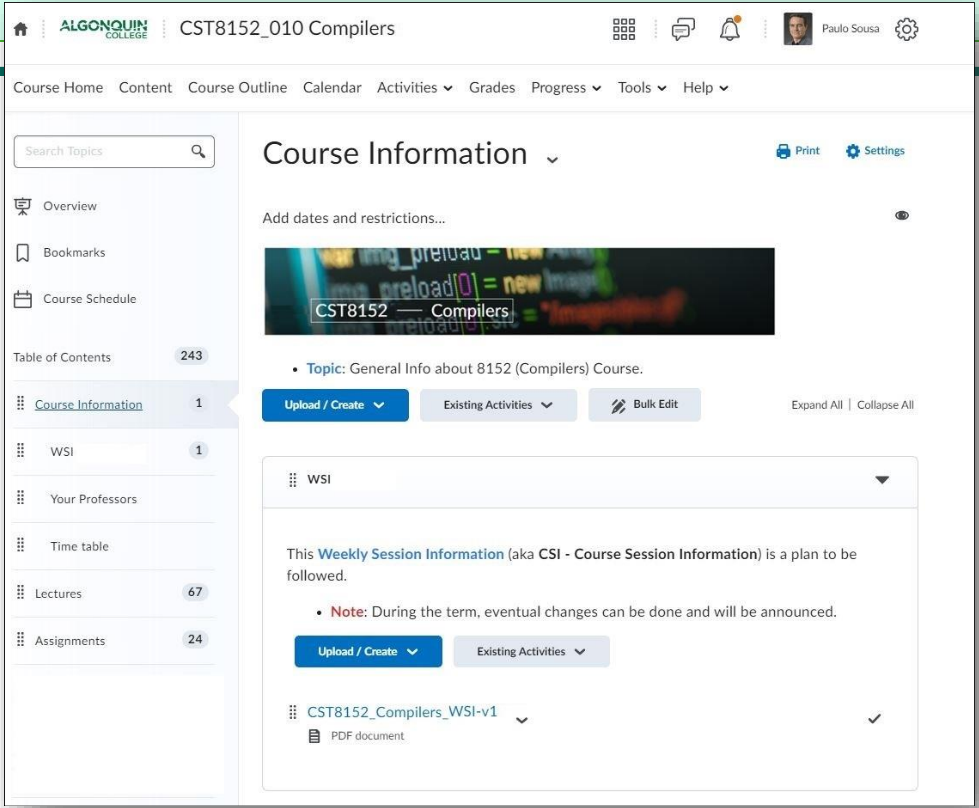
Recommendation!

Software Recommended:
MS Visual Studio 2022 Community Edition
or any ANSI C compliant compiler.

Remember that you need to use **ANSI C.**

WSI (CSI)

- Most important resource:
  - Brightspace – Course Information



- Topic: General Info about 8152 (Compilers) Course.

This Weekly Session Information (aka **CSI - Course Session Information**) is a plan to be followed.

- Note: During the term, eventual changes can be done and will be announced.

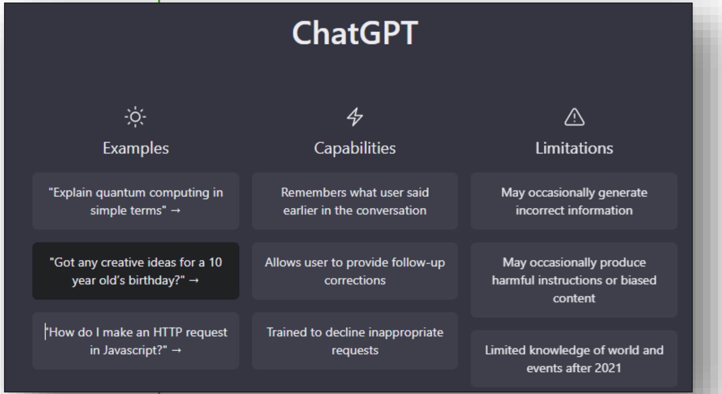
What about… ChatGPT

- **ChatGPT** was fine-tuned on top of GPT-3.5 using **supervised learning** as well as **reinforcement learning**.

Both approaches used human trainers to improve the model's performance. It uses the Proximal Policy Optimization (PPO) algorithms present a cost-effective benefit to trust region policy optimization algorithms and the models were trained in collaboration with Microsoft on their Azure supercomputing infrastructure.

- ➢ Basic Chat (Sep 1st 2023): https://openai.com/blog/chatgpt/

OpenAI ChatGPT

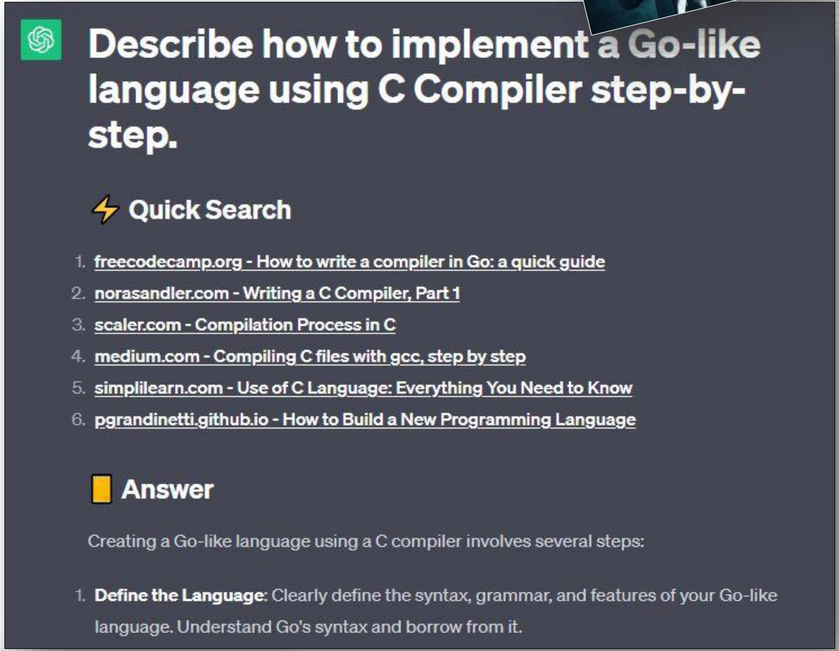


ChatGPT

Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

1.4. But…

- **Trying…**



Describe how to implement a Go-like language using C Compiler step-by-step.

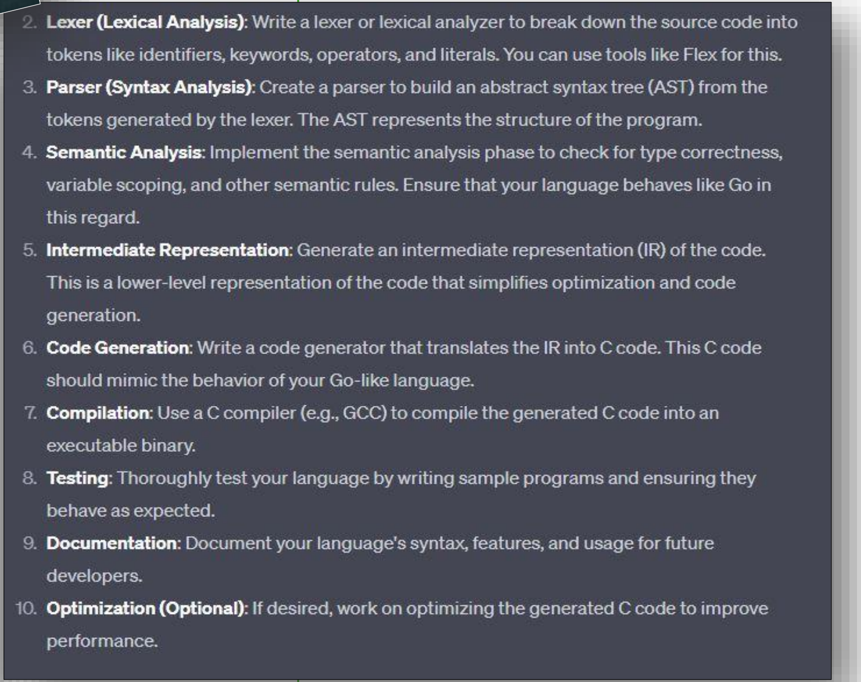
⚡ Quick Search

1. freecodecamp.org - How to write a compiler in Go: a quick guide
2. norasandler.com - Writing a C Compiler, Part 1
3. scaler.com - Compilation Process in C
4. medium.com - Compiling C files with gcc, step by step
5. simplilearn.com - Use of C Language: Everything You Need to Know
6. pgrandinetti.github.io - How to Build a New Programming Language

Answer

Creating a Go-like language using a C compiler involves several steps:

1. **Define the Language**: Clearly define the syntax, grammar, and features of your Go-like language. Understand Go's syntax and borrow from it.



2. **Lexer (Lexical Analysis)**: Write a lexer or lexical analyzer to break down the source code into tokens like identifiers, keywords, operators, and literals. You can use tools like Flex for this.
3. **Parser (Syntax Analysis)**: Create a parser to build an abstract syntax tree (AST) from the tokens generated by the lexer. The AST represents the structure of the program.
4. **Semantic Analysis**: Implement the semantic analysis phase to check for type correctness, variable scoping, and other semantic rules. Ensure that your language behaves like Go in this regard.
5. **Intermediate Representation**: Generate an intermediate representation (IR) of the code. This is a lower-level representation of the code that simplifies optimization and code generation.
6. **Code Generation**: Write a code generator that translates the IR into C code. This C code should mimic the behavior of your Go-like language.
7. **Compilation**: Use a C compiler (e.g., GCC) to compile the generated C code into an executable binary.
8. **Testing**: Thoroughly test your language by writing sample programs and ensuring they behave as expected.
9. **Documentation**: Document your language's syntax, features, and usage for future developers.
10. **Optimization (Optional)**: If desired, work on optimizing the generated C code to improve performance.

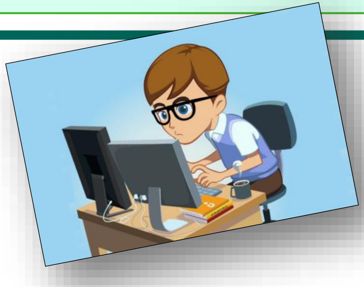
OpenAI ChatGPT

Your Voice, your Rights…

- Any Problems:

You can contact me by email or on MS Teams (Paulo Sousa – sousap@algonquincollege.com).

Or, if I am not doing well my role, you can contact the coordinator (CET), prof. Howard Rosenblum (rosenbh@algonquincollege.com) or our Chair, prof. Sandra Brancatelli (brancas@algonquincollege.com).



Computer Programming and Analysis
(Co-op and Non-Co-op Options)

Final Message…

Final Message…

Enjoy our course and season…

Compilers – Week 1

Thank you for your attention!

Contact: sousap@algonquincollege.com

ALGONQUIN COLLEGE