

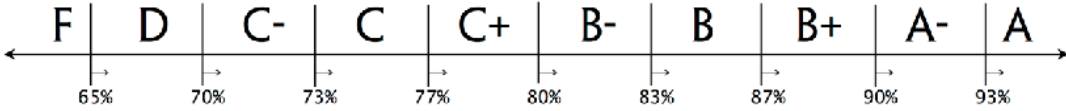
# Compilers

CMPT 432 • Spring 2024

## -Background

When and where	Class Mondays 8AM to 10:45AM in Music 3202. Labs Thursdays 8AM or online.	
Texts	<i>Crafting a Compiler</i> by Fischer, Cytron, and Leblanc, Jr. - "CaC" published by Addison Wesley in 2010. ISBN 978-0-13-606705-4 <i>Compilers: Principles, Techniques, and Tools</i> by Aho, Lam, Sethi, and Ullman - "Dragon" published by Addison-Wesley in 2007. ISBN 0-321-48681-1	
Web	<a href="https://www.labouseur.com/courses/compilers">https://www.labouseur.com/courses/compilers</a>	
Instructor	Alan G. Labouseur Hancock 3007	Alan.Labouseur@Marist.edu Office hours are posted.

## -Grading

Letter Grades				
You can earn up to 1000 points, broken down as follows:	Project One and labs	10.0%	100 points	[1, 2]
	Project Two and labs	10.0%	100 points	[1, 2]
	Project Three and labs	10.0%	100 points	[1, 2]
	Project Four and labs	15.0%	150 points	[1, 2]
	Mid-term Exam	25.0%	250 points	[1, 2]
	Final Exam	25.0%	250 points	[1, 2]
	Attendance and Participation	2.5%	25 points for quality and quantity	[1]
	Laziness and Whining	2.5%	25 points for not (lazy or whining)	[1]

## -Themes, Objectives, and Assessment

Assessment methods include assignments, quizzes, exams, discussions, presentations, peer review, and projects.	In this course, I hope you will ...
[References] refer to Department of Computing Technology Goals available at Labouseur.com	<ul style="list-style-type: none"><li>• gain and demonstrate an understanding of the fundamental areas of compiler architecture: front end, intermediate representation, and the back end [1, 2];</li><li>• gain and demonstrate an understanding of context-free grammars and their use [2];</li><li>• gain and demonstrate an understanding of the techniques for scanning (lexical analysis), parsing a grammar, translation, and simple code generation [1, 2];</li><li>• embrace the opportunity to develop a complex system over the course of the semester where you have to either live with your prior mistakes and shortcuts or go back and fix them. (Either will teach valuable lessons.) [1, 2]</li><li>• learn that developing the software is only half the battle, debugging and testing are critical skills for a talented professional, and skills that will be valuable. [1, 2]</li><li>• gain and demonstrate an understanding that the chasm between programs that work once and programs that work every time is ridiculously huge [1, 2];</li><li>• enhance your continuing education skills. Capable problem solvers never stop learning. You will get practice in finding answers for yourself. [1, 2]</li></ul>
	Finally, this class and its project are popular interview topics. I want you to be so awesome that the company hires you on the spot and sends you home with a Brinks truck full of cash and video games, including Portal 3.

# Compilers

CMPT 432 • Spring 2024

## -Schedule

#	Week	Due	S	CaC	Dragon	Topics
0	22-Jan	Lab 0	I	1 10.1.2	1	Introduction • Demo • Overview (CFG <sup>2</sup> ) • Brief history • Classification of programming languages • Compilation phases • Design considerations
1	29-Jan	Lab 1	L	3	3	Lexical Analysis • Tokens • Symbol Lists Regular Expressions • State Machines • Finite Automata in general • DFAs
2	5-Feb	Lab 2	L	3	3	Transition tables for DFAs • NFAs • Regular Expressions to NFAs Chaining NFAs together • Turning NFAs into DFAs
3	12-Feb	<b>Project One</b>	P1	4.1-4 5.1-3	2.7, 2.8.2 4.2	Context free grammars, derivations, and reductions • Syntax trees Top-down parsing • Recursive descent parsers
4	19-Feb	Lab 3	P1 P1½	7.1	2.7, 2.8.2 4.4.1 5.3.1	More grammars and derivations • Recursive descent parsers Building a CST • Implementing trees
5	26-Feb	Lab 4	P2	4.5 5.9	4.1 4.4.2	More on building CSTs • Trees First and Follow sets • Error handling and recovery during parsing
6	4-Mar	Lab 5 CST	-	—	—	<b>Mid-term Exam</b> at 8AM in HC 2023 One-page study sheet permitted. Some restrictions apply.
7	<del>11-Mar</del>	<b>Project Two</b>	-	—	—	<i>No class meeting, Spring Break.</i> But Project Two is due on this date.
8	18-Mar	Lab 5 AST	P3 SA1	8.1-3 7.1 7.3-7	2.7 6.3	Variables and types • Static and Dynamic Scope • Symbol Tables • Abstract Syntax Tree patterns in the CST • Building an AST from patterns in the CST
9	25-Mar	Lab 6	SA2 SA3	2.7 7.3-7 8.1-3, 9.1	2.8.3 6.3, 6.5	Source code → CST → AST • Checking scope and building a symbol table Checking types • Type systems
A	<del>1-Apr</del>	Lab 7	-	—	—	<i>No class meeting, it's the final few hours of Easter Break.</i> (Work on Lab 7 because it will help you with Project Three)
B	8-Apr	<b>Project Three</b>	SA3 CG	7.3-7 8.1-3, 9.1	6.3, 6.5 7.1, 7.4 8.1, 8.3.1	More on Type systems (but not <i>moron</i> type systems, that's Python) 6502a op codes • Code Generation • Runtime environment
C	15-Apr	Lab 8	CG	12.1 13.1-2	6.6-7 7.1, 7.4 8.1, 8.3.1	Static allocation • AST → 6502a op codes Heap management • more AST → 6502a code generation
D	22-Apr	Lab 9	-	—	—	<i>Catch-up Class</i>
E	29-Apr	-	P4	5.5 6.1-2	2.4.5 4.3, 4.5-6 4.8	LL(1) analysis • Grammar ambiguity • Associativity and Precedence Left recursion • Left factoring • Bottom-up (LR) parsing
F	6-May	<b>Project Four</b>	-	—	—	<b>Final Exam</b> at 8AM in HC 2023 One-page study sheet permitted. Some restrictions apply.