Compilers

CMPT 432 • Spring 2018

Background

When and where

Tuesday 8am through 10:45am in LT 21. Labs online and Fridays 8am to 9am

Suggested Texts

Crafting a Compiler by Fischer, Cytron, and Leblanc, Jr. - "CaC"
published by Addison Wesley in 2010. ISBN 978-0-13-606705-4

Compilers: Principles, Techniques, and Tools by Aho, Lam, Sethi, and Ullman - "Dragon"
published by Addison-Wesley in 2007. ISBN 0-321-48681-1

Web

http://www.labouseur.com/courses/compilers

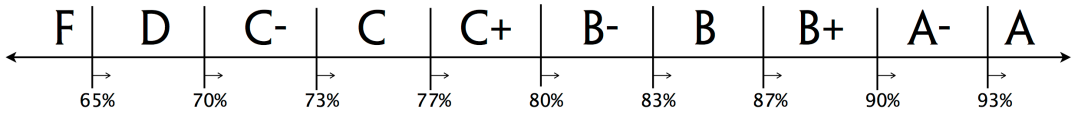
Instructor

Alan G. Labouseur
Hancock 3***007***
Office hours are posted.

Alan.Labouseur@Marist.edu
845-575-3832 *Marist phone*
845-440-1102 *home office phone*

Grading

Letter Grades



You can earn up to 1000 points over the course of the semester, broken down as follows: (These weights are subject to minor variation.)

Project One and labs	10.0%	100 points	[1, 2]
Project Two and labs	10.0%	100 points	[1, 2]
Project Three and labs	15.0%	150 points	[1, 2]
Project Four and labs	20.0%	200 points	[1, 2]
Mid-term Exam	20.0%	200 points	[1, 2]
Final Exam	20.0%	200 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]

Objectives and Assessment

Assessment methods include assignments, quizzes, exams, discussions, presentations, peer review, and projects.

[References] refer to Department of Computing Technology Goals available at http://www.labouseur.com/courses/goals.pdf

In this course, I hope that you will . . .

- gain and demonstrate an understanding of the fundamental areas of compiler architecture: front end, intermediate representation, and the back end [1, 2];
- gain and demonstrate an understanding of context-free grammars and their use [2];
- gain and demonstrate an understanding of the techniques for scanning (lexical analysis), parsing a grammar, translation, and simple code generation [1, 2];
- 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 a valuable lesson.) [1, 2]
- 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]
- enhance your continuing education skills. Capable problem solvers never stop learning. You will get practice in finding answers for yourself. Additionally, preparation and presentation of the projects, as well as participation in class discussions and assignments, requires at least a little research, so there's that. [1, 2]

Compilers

CMPT 432 • Spring 2018

Proposed Schedule

#	Week	Due	S	CaC	Dragon	Topics
0	16-Jan	Lab 0	I	1 10.1.2	1	Introduction • Demo • Overview (CFG²) • Brief history • Classification of programming languages • Compilation phases • Design considerations
1	23-Jan	Lab 1	L	3	3	Lexical Analysis • Tokens • Symbol Lists Regular Expressions • State Machines • Finite Automata in general • DFAs
2	30-Jan	Lab 2	L	3	3	Transition tables for DFAs • NFAs • Regular Expressions to NFAs Chaining NFAs together • Turning NFAs into DFAs
3	6-Feb	***Project One***	P1	4.1-4 5.1-3	2.7, 2.8.2 4.2 4.4.1	Context free grammars, derivations, and reductions • Syntax trees Top-down parsing • Recursive descent parsers • Symbol List ➛ Symbol Table
4	13-Feb	Lab 3	P1 P2	7.1	2.7, 2.8.2 4.4.1 5.3.1	Reflect on Project One • More grammars, derivations, and recursive descent parsers • Symbol Tables • Building a CST • Implementing trees
5	20-Feb	Lab 4	P3	4.5 5.9	4.4.2	First and Follow sets • Error handling and recovery during parsing
6	27-Feb	Lab 6 partial	-	—	—	**Mid-term Exam** in class One-page study sheet permitted. Some restrictions apply.
7	6-Mar	***Project Two***	- P2	8.1-3	2.7 6.3	Meditate on the mid-term exam Variables and types • Static and Dynamic Scope
-	13-Mar	—	-	—	—	*No Class Meeting: Spring Break*
8	20-Mar	Lab 5	SA1 SA2	7.1 7.3-7	6.3	Consider Project Two • Review scope • Abstract syntax trees (AST) • AST patterns in the CST • Building an AST from patterns in the CST
9	27-Mar	rest of Lab 6, Lab 7	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 Type systems • Checking types • Source code ↦ AST
A	3-Apr	***Project Three***	SA3 CG	2.7.2 8.1-3	2.8.3 6.5	More Source code ↦ CST ↦ AST • Checking scope and type 6502a op codes • Introduction to code generation
B	10-Apr	Lab 8	CG	12.1 13.1-2	6.6-7 7.1, 7.4 8.1, 8.3.1	Discuss Project Three • More 6502a op codes and code generation • Runtime environment • Static allocation • Heap management • AST ↦ 6502a op codes
C	17-Apr	—	-	—	—	*No Class Meeting: Time-wasting faculty assessment day*
D	24-Apr	—	P4	5.5 6.1-2	2.4.5 4.5-6 4.8	LL(1) analysis • Grammar ambiguity • Associativity and Precedence • Left recursion • Left factoring • Bottom-up (LR) parsing with Shift-Reduce
E	1-May	Lab 9	-	—	—	**Final Exam** in class One-page study sheet permitted. Some restrictions apply.
F	May 9 10:30am	***Project Four***	-	—	—	Show off your awesome compiler.