

Compilers

CMPT 432 • Spring 2022

-Background

| | | |
|----------------|--|--|
| When and where | Class Mondays 8AM to 10:45AM in Hancock 1021. Labs Tuesdays 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 | https://www.labouseur.com/courses/compilers and iLearn | |
| Instructor | Alan G. Labouseur Hancock 3007 <i>Office hours are posted.</i> | Alan.Labouseur@Marist.edu 845-575-3832 <i>Marist</i> 845-440-1102 <i>home office</i> |

-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 | 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] | | | | | | |

-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">• 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]• gain and demonstrate an understanding that the chasm between programs that work once and programs that work every time is ridiculously huge [1, 2];• enhance your continuing education skills. Capable problem solvers never stop learning. You will get practice in finding answers for yourself. [1, 2] |
| | 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 2022

–Planned Schedule

| # | Week | Due | S | CaC | Dragon | Topics |
|---|-------------------|----------------------|------------|----------------------------|---------------------------------|---|
| 0 | 24-Jan | Lab 0 | I | 1 10.1.2 | 1 | Introduction • Demo • Overview (CFG ²) • Brief history • Classification of programming languages • Compilation phases • Design considerations |
| 1 | 31-Jan | Lab 1 | L | 3 | 3 | Lexical Analysis • Tokens • Symbol Lists Regular Expressions • State Machines • Finite Automata in general • DFAs |
| 2 | 7-Feb | Lab 2 | L | 3 | 3 | Transition tables for DFAs • NFAs • Regular Expressions to NFAs Chaining NFAs together • Turning NFAs into DFAs |
| 3 | 14-Feb | Project One | 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 | 21-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 | 28-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 | 7-Mar | Lab 5 CST | - | — | — | Mid-term Exam in HC 2023 One-page study sheet permitted. Some restrictions apply. |
| 7 | 14-Mar | Project Two | - | — | — | <i>No class meeting, Spring Break.</i> But Project Two is due on the 14th. |
| 8 | 21-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 | 28-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 | 4-Apr | Lab 7 | SA3 | 7.3-7 8.1-3, 9.1 | 6.3, 6.5 | More on Type systems (and moron type systems) Catch-up and review • Project Three help |
| B | 11-Apr | Project Three | CG | 12.1 13.1-2 | 6.6-7 7.1, 7.4 8.1, 8.3.1 | 6502a op codes • Code Generation • Runtime environment Static allocation • AST ⇨ 6502a op codes |
| C | 18-Apr | - | - | — | — | <i>No class meeting, Easter Break.</i> |
| D | 25-Apr | Lab 8 | CG | 12.1 13.1-2 | 6.6-7 7.1, 7.4 8.1, 8.3.1 | Heap management • more AST ⇨ 6502a op codes |
| E | 2-May | Lab 9 | 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 | 12-May | Project Four | - | — | — | Final Exam Thursday, May 12th at 8AM in HC 2023 One-page study sheet permitted. Some restrictions apply. |