CSC-490: Compiler Design

Aaron Cass

Spring 2024

Course Description

This course will be a hands-on introduction to compiler design, focused on the development of the Charge programming language. Students will participate in weekly scrums, peer coding, and project management to create starter code for future classes.

Course Objectives

By the end of this course, students will be able to:

- Understand the fundamentals of compiler design and apply them to the development of a compiled programming language.
- Contribute effectively to a large software development project in a team environment.
- Implement best practices in coding, including clear commenting, frequent compilation, and effective version control.
- Develop robust documentation, unit tests, and regression tests to ensure software quality and reliability.

Course Schedule

Weeks 1 - 2: Initial Development: Each person will have completed 2 issues, especially on lexing and parsing.

Week 3 - 4: Core Compiler Functionalities: Each person will have completed 5 issues, especially on symbol table stack generation, semantic analysis, and module structuring. Students will also split into three teams: Frontend, Backend, and Testing.

Week 5 - 9: Features and Testing: Students will develop LLVM tools, optimization techniques, unit tests, and regression testing. Finalize documentation and testing per team.

Week 10: Final Development: Students finalize primary objectives and prepare for project submission.

Week 11: Project Presentations and Review: Submission of Charge compiler project.

Textbooks

- Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. Compilers: Principles, Techniques, & Tools. Pearson Education, second edition, 2007.
- John Cocke and J.T. Schwartz, Programming Languages and their Compilers, Preliminary Notes, 2nd revised edition, Courant Institute, New York, 1970.

Grading

- **100%**: TBD