CSC 4351: Compiler Construction

Syllabus

Spring 2020

Course Summary

Program language structures, translation, loading, execution, and storage allocation; compilation of simple expressions and statements; organization of compiler including compile-time and run-time symbol tables, lexical scan, syntax scan, object code generation, error diagnostics, object code optimization techniques, and overall design; use of compiler writing languages and bootstrapping.

Prerequisite

According to the course listing, the prerequisites are:

CSC 4101 or equivalent.

Office Hours

Gerald Baumgartner: MW 9:30-11:00am, PFT 3272D, Tel: 578–2191, Email: gb
Monazil Chowdhury: Email: mchow15

Other office hours by appointment (recommended).

Important Dates

- Tailgating: Jan 13
- Martin Luther King Day: Jan 20
- Mardi Gras: Feb 24
- Midterm: Mon, Mar 2
- Spring break: Mar 22–29
- Final: Tue, May 5, 12:30–2:30pm

Both exams are comprehensive.

Reading

Projects

There will be six programming assignments. Each will take about two to two and a half weeks.

A penalty of 10% will be assessed for each day a project is late up to a maximum of 30% after which the project will not be accepted. The final project may not be turned in later than Saturday midnight after the last day of classes.

Homeworks

There will be four short homework assignments. A penalty of 20% will be assessed if a homework is submitted at the beginning of the next class period. Homeworks later than that will not be accepted.

Annotated Bibliography

Graduate students or honors students who want to honorize this course must submit an annotated bibliography on a compiler topic.

Grading

- Projects: 42%
- Homeworks: 8%
- Bibliography: 3%
- Midterm: 20%
- Final: 30%

Topics

This list of topics is an ordering of subjects that will be covered with the corresponding book chapters and assignments. Each topic will roughly take a week and a half, with a bit less time for topics 1 and 9.

1. Compiler Structure: Ch 1
2. Lexical Analysis: Ch 2, Proj 1, HW 1
3. Parsing and Abstract Syntax: Ch 3 & 4, Proj 2, HW 2
4. Semantic Analysis: Ch 5, Proj 3
Due Dates and Grading

The time allotted for each homework assignment will be at least one week. The penalty for turning in late is 20 percent by the next class period. After that homeworks will not be accepted. I am planning to experiment with using Gradescope for simplifying the grading of exams and homeworks. If we get Gradescope set up for homework submissions, then you would upload scans of your homework to Gradescope. Otherwise, homeworks will be submitted on paper in class.

Since the time needed for finishing a programming assignment is harder to estimate and to allow fixing severe bugs that show up close to the deadline, programming assignments can be submitted up to three days after the official deadline. The last programming assignment cannot be submitted later than by Saturday midnight after the last day of classes. For each day past the deadline, a penalty of 10 percent will be incurred. Programming assignments will be submitted electronically and will be due at midnight. We will not accept submissions more than three days after the due date.

Without prior arrangements in case of extenuating circumstances, submission of homeworks and project past the late deadline is not allowed and such work will not be graded and you will receive no credit. It is your responsibility to make sure that you have completed your work with enough time to submit your materials. For projects, make sure that you verify that your code was submitted correctly.

Grading disputes can be submitted in writing with accompanying documentation, or in person during regular office hours. It is course policy that whoever graded the work will be responsible for handling disputes. In general, I will grade the midterm exam and the final exam. The grader will grade the homeworks and the programming assignments. Grades become final one week after a homework, project, or exam is handed back. This should leave ample time for resolving grading disputes. This rule will be strictly enforced, groveling for a point here and a point there at the end of the quarter will do you no good!

Homework Standards

All written work submitted must carry the student’s name and must be reasonably neat and well organized. Any work that cannot be read easily will score zero points. A reasonable standard of English expression and grammar is also required. The same requirements apply to exams.

Programming Standards

The algorithm used must be essentially correct. Obviously, the program should compile and run. Because of the complexity of some of the programs, no credit may be given for a program that doesn’t run. If a program throws a run-time exception, only partial credit will be given.
Since projects build on top of previous projects, it is very important to get each submission to run without run-time exceptions and to structure the program so it can be easily extended.

I expect your work to exhibit high standards of programming style and layout, reflecting your expertise as a computer professional. Poor style and documentation may result in up to 10% deducted.

**Honesty**

I will treat you as professionals, and you should plan on conducting yourself as such. This course presents many important concepts you will need throughout your career as a computing professional, so it is important that each student do all the assignments and projects and learn the material.

There will be several homework assignments and programming projects. You are free to discuss these assignments with others. However, the programs and homework solutions you submit are to be developed by yourself (or by your team). *Cheating is a very serious offense and will not be tolerated.* The grader or I will use tools for detecting cheating on programming assignments. Reverse-engineering the reference implementation or supplying others with material is also against this rule. Similarly, making homework solutions or code available publicly on a web site, such as Github, is also considered a cheating offense. The policy is that the supplier and receiver of information will both be reported to the Dean of Students.

Save all handwritten notes or printouts you generate as you work on a project and keep them until the end of the semester so as to protect yourself in the event that someone “borrows” your program, or the version you submit is mislaid.

**Computer Account Security and Use**

To help others resist the temptation of using your work, you should maintain proper security on your computer account. Especially, keep your password from others and do not alter the protection on any of your files. To give others access to your account or files or printouts of your programs is the same as giving them the information directly and will be dealt with accordingly. Any trouble with computer accounts should be referred to an instructor as soon as possible.

When a program has been submitted electronically, you should maintain an *unedited* version of what you submitted (with the correct date stamp) until after that program has been graded. It is also beneficial to use version control software such as git or CVS to keep track of all versions of files so you can revert back to an old version if necessary.