

CSC 4351

Compiler Construction

Spring 2018

Syllabus

Gerald Baumgartner

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

Who	Where	Phone	E-Mail	When
Gerald Baumgartner	Patrick Taylor 3272D	578-2191	gb	MW 10:00-11:30am
TBA			TBA	

Other office hours by appointment (recommended).

Important Dates

- Martin Luther King Day: Jan 15
- Mardi Gras: Feb 12
- Midterm: Mon, Mar 5
- Spring break: Mar 25-Apr 1
- Final: Thu, May 3, 7:30-9:30am

Both exams are comprehensive.

Reading

- Andrew W. Appel, *Modern Compiler Implementation in Java*. Cambridge University Press, 1998 or 2002 (required).
- Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Monica S. Lam, *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 2006 (optional).
- Thomas W. Parsons, *Introduction to Compiler Construction*. Computer Science Press, 1992 (optional).
- <http://docs.oracle.com/javase/tutorial/>
- <http://docs.oracle.com/javase/specs/>
- <http://docs.oracle.com/javase/8/docs/api/>
- Mailing list csc4351@csc.lsu.edu
- Course web page <http://www.csc.lsu.edu/~gb/csc4351/> and Moodle

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

Labs	42%
Homeworks	8%
Bibliography	3%
Midterm	20%
Final	30%

Topics

This list of topics is primarily an ordering of subjects to be covered. Each topic will roughly take a week and a half, with a bit less time for topics 1 and 11.

	Topic	Reading	Due
1	Compiler Structure	Ch 1	
2	Lexical Analysis	Ch 2	
3	Lexical Analysis, Parsing	Ch 3	Lab 1
4	Parsing	Ch 3	HW 1
5	Abstract Syntax, Semantic Analysis	Ch 4	Lab 2
6	Semantic Analysis	Ch 5	HW 2, Lab 3
7	Review, Activation Records	Ch 6	Midterm
8	Activation Records	Ch 6	Lab 4
9	Intermediate Code, Canonicalization	Ch 7, 8	HW 3
10	Instruction Selection	Ch 9	Lab 5
11	Liveness Analysis, Register Allocation	Ch 10, 11, 12	HW 4, Lab 6

Due Dates and Grading

The time allotted for each homework assignment will be made quite generous, as such, the penalty for turning in late is high: 15% per day up to a maximum of 30%. If you turn in a homework on a weekend, write on the submission date and time and slide it under my door. I usually stop by my office on weekends. Predating a late submission is considered a cheating offense. On weekends, submission by Saturday 5pm will incur a penalty of 15%; submission by the class period on Monday will incur a penalty of 30%.

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. Labs will be due at 12am midnight. The submission facility will reject your submissions three days after the due date.

Without prior arrangements in case of extenuating circumstances, submission of homeworks and labs 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.

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, lab, 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 can be given for a program that doesn't run. If a program dumps core (throws a run-time exception), only partial credit will be given.

Since labs build on top of previous labs, it is very important to get each submission to run without core dumps 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. The policy is that the supplier and receiver of information will both be reported to the Dean of Students.

Save all handwritten notes and printouts you generate as you work on a project and keep them until the end of the quarter so as to protect yourself in the event that someone "borrows" your program, or the version you submit is mislaid. For your protection, cases of missing output should be *immediately* reported.

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.