

Compiler Design

CSCI 425

Your homework is here (write this down):

`https://cs.mcprogramming.com/`

follow the course link, and click on "Schedule" in the Wiki tab bar.

Lexing and Scanning

Context Free Grammars

LL Parsing (Recursive Descent)

LR Parsing (Bottom-Up)

Parse Tree \rightarrow Abstract Syntax Tree \rightarrow Target

Formal Learning Groups

(“...and what are these?” you wonder)

What Your Peers Think

1. What aspects of instruction did you find effective for promoting your learning in this course?

I have grown to like the Learning Groups, because we can sometimes learn better from our peers than the Instructor,

I have grown to like the learning groups, because we can sometimes learn better from our peers than the instructor.

What Your Peers Think

1. What aspects of instruction did you find effective for promoting your learning in this course?

I enjoyed the learning groups and I believe they helped me pay attention better as well as provided incentive to keep up with work so as to not let my group down.

I enjoyed the learning groups and I believe they helped me pay attention better as well as provided incentive to keep up with the work so as to not let my group down.

Minor Course Details :)

Syllabus & Book & Grading & `alamode` Assignments

Canvas **will not** be used (the course has been published with a lone announcement saying (guess what): *Canvas won't be used.*

Announcements via Email to your
`mines.edu` address.

Q & A

Got some?

Compilers, compilers, everywhere...

We are all CS people, and we all use compilers all over the place. LLVM, L^AT_EX, `grep(1)`, `vi/perl/awk` search and replace, Markdown's `pandoc`, `javac`, `python -OO ...`

Take two minutes and reflect a little bit:

- ▶ Think of two of your favorite features of compilers you use. Be **careful**! These are things like *very helpful* warning or error messages, or particular options and invocations (e.g.: `-syntax-only`). These are not features of an editor or IDE! We want features or properties of **compilers**.
- ▶ What are two critical tasks you expect any compiler to always be able to do, and always do correctly.

Introducing Yourself

Name (First and **Last**)

Major

Complete the sentence “(I | I’m a) ??????”