

COMP 520 Compiler Design **Group Milestone #1**

Scanner, Parser and Pretty Printer for GoLite....

Due: Sunday, February 26

The purpose of this milestone is to get the first phase of your project compiler completed.

Question 1: *Programs* (10 points)

The first question for this milestone will get you started on building a complete test suite for your compiler. To this end, you must:

1. Develop 2 correct example program per team member (i.e. either 4 or 6 programs, depending on the number of people in your team). These programs must compute something useful. We will make a library of these programs for further testing.
2. Develop 10 invalid programs per team member (i.e. either 20 or 30 programs, depending on the number of people in your team). These programs should cover a large number of test areas.

Note: You should test your compiler on *many* more programs, especially invalid ones for errors, but we will not grade these. A complete test suite could involve several times more programs. Using an automated test script as was used in the minilang assignments will allow you to quickly and completely re-test your compiler after making possibly breaking changes.

Question 2: *Scanner, Parser and Pretty Printer* (30 points)

Implement the scanner and parser to generate an AST, and a pretty printer from the AST in the language of your choice.

Given a syntactically correct input program of the name `foo.go`, your compiler should write the pretty print to file `foo.pretty.go`. This pretty-printed file should be parsable by your compiler, in particular, check the invariant we saw in class:

$$\text{pretty}(\text{parse}(P)) \equiv \text{pretty}(\text{parse}(\text{pretty}(\text{parse}(P))))$$

Your front-end should handle errors in a user-friendly way. You need to only catch the first error and then quit, but you should try to give a reasonable error message. Error messages should be displayed on a single line (this helps the T.A. automate testing) and be sent to `stderr`. As in the minilang assignments, we will be using test scripts to evaluate your submissions.

Question 3: *Design Decisions and Team Work* (10 points)

Briefly discuss the design decisions you took in the design and implementation of your scanner/parser/pretty-printer. If there are parsing issues that you are deferring to a weeding phase, please document them here (you may also want to implement these weeding passes as part of this milestone). Also include in this discussion:

- the rationale of the implementation tools and language that you chose
- summarize how your team is organized and what each team member contributed

Note: You should also keep notes on each phase, as this will help you generate the final project report.

What to hand in

You will be developing your project in your team's github repository. At each milestone you will create a tag before the due date, and the TAs will review the code associated with that tag. For this first milestone you should create a tag called *milestone1*. Information about creating git tags can be found at: <http://git-scm.com/book/en/v2/Git-Basics-Tagging>.

Your project should be kept in the following format:

```
/  
  README      (Your group names, student IDs, relevant info and  
               instructions for each milestone (just add information  
               as you finish each milestone. Make it easy for the TAs to  
               grade your milestone!)  
  programs/  
    valid/    (your valid programs)  
    invalid/  (your invalid programs)  
  src/       (the source code and build files. You must use some sort of  
               automatic build system like Makefile or ant)  
  doc/       (design documents, the answer for question 3 should be in a  
               file called milestone1.pdf)  
  build.sh   (a build script to compiler your compiler)  
  run.sh     (a run script that when invoked as "run derp.go" will run  
               your compiler)
```