COMP 520 Compiler Design
Group Milestone #1

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

Due: Friday, February 26

Overview:

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

Question 1: *Programs* (10 points)

For GoLite teams, develop 1 example program per team member (i.e. either 2 or 3 programs, depending on the number of people in your team). For OncoTime teams develop 2 example programs per team member (either 4 or 6 programs, depending on the number of people on your team).

These programs should compute something useful. We will make a library of these programs for further testing.

You should test your compiler on many more programs, especially invalid ones for errors, but we will not grade these.

All the valid and invalid programs you submit will be put together into a big test suite that we will use to grade your scanner and parser.

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

For either GoLite or OncoTime, implement the scanner and parser to generate an AST, and a pretty printer from the AST.

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

$$pretty(parse(P)) \equiv pretty(parse(pretty(parse(P))))$$

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.

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.

Also summarize how your team is organized and what each team member contributed to this milestone.

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 for testing)
    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)
```