Lab 3

COEN 175 Compilers

Overview For Lab 2

Goal

- Write recursive-descent parser for Simple C that will print out the operation order of a given input program

Submission

- Submit a tarball of your cpps and make files in folder called phase2
- Due Date: Sunday January 23rd

Main Objectives

* You will be given a working solution for phase 1
- Last Week:
  - Disambiguate your grammar by “hand”
  - Modify `lexer.cpp`
  - Create `parser.cpp`
- This Week:
  - Left factor translation unit
  - Implement the rest of the grammar

Finishing up Parser

- Function per grammar section similar to expression
- Statements
  - No clear way to check end based on statements definition
  - Any list of statements always ends with a }
- Specifiers
  - Have your specifier function return what specifier it matched
- Pointers
  - Infinite pointers in theory
  - Write a single function to match them all
- Declarations

Left Factoring Translation Unit

- Combine
  - Type-definition
  - Function-definition
  - Global-declaration,
  - Global-declarator-list.
- This should be your top level function that is called in your main
- Check Lecture 5 slides

Testing your code

Examples provided by us at **`/scratch/coen175`**
Run the following commands:

```
$ make clean all
$ ./scc < /scratch/coen175/[file].c | diff - /scratch/coen175/[file].out
```

[file] determined by what part you're testing

- expression() ⇒ exp1 and exp2
- statement() ⇒ statement1
- declaration() ⇒ declaration1

Use **`/scratch/coen175/exampleMain.cpp`** to see how you should write your main for each step

Tips

- Check the lecture slides, they are VERY helpful
- Check your cases with parentheses
- Thoroughly test your code with your own examples
  - The provided examples will not test everything
- **READ THE SYNTAX RULES CAREFULLY**