CPSC 4600 Project Language (PL) Compiler

Spring 2019

Roderick MacCrimmon (single person group, responsible for whole project)

Estimated time spent on project: 10-15 hours (2 weekends - did not really keep track)

Files & Directory Layout

(excluding CMake build files)

- README.pdf
- technical_doc.pdf
- **include/** - header files
  - compiler.h
  - scanner.h
  - symbol_table.h
  - symbol.h
  - token.h
- **src/** - implementation files
  - main.cpp - contains main function for plc
  - compiler.cpp
  - scanner.cpp
  - symbol_table.cpp
  - token.cpp
- **test/**
  - catch.hpp - CATCH2 unit testing library header
  - main.cpp - defines main needed for running unit tests
  - test_scanner.cpp - unit tests
  - **src_files/** - PL source code used for testing
    - valid_test_program.pl
    - error_test_program.pl
    - invalid
    - token_types
- **docs/**
  - README.md

- technical_layout.tex

Build Instructions

This project uses CMake to automatically generate makefiles. It has been tested to ensure it builds and runs on the linux lab computers without any compilation or runtime errors.

From the project root directory run the following commands:

```
mkdir build
cd build
cmake ..
make
```

This will produce two separate executables - one for the main compiler at build/src/plc, and a second for the automated unit tests at build/test/run_tests

Usage Instructions

The compiler can be run with

```
./plc src_file [-o output_file]
```

The source file must always be provided as the first argument. The output file is optional and if none is provided, output will be written to scanner.out

Two example files are provided to see the output at test/src_files/valid_test_program.pl and test/src_files/error_test_program.pl

The unit tests can be run from the project root directory with

```
./build/test/run_tests
```

Because the input filepaths are currently hard-coded, this must be run from the root directory.