

COSC 4P75 Assignment 2

Fall 2011/12

Due: Nov. 7 @ 12:00 noon.

Parser

Write and test the syntactic analyzer (parser) for the project language (<http://www.cosc.brocku.ca/Offerings/4P75/Syntax.pdf>). The implementation language is Java. Make appropriate use of Java libraries (except, of course, syntactic analysis facilities) and object-oriented techniques.

The parser is intended for a single-pass compiler and should use the lexical analyzer from assignment 1 as an input filter providing single symbol (token) lookahead. It should use the techniques discussed in class and provide appropriate error detection, reporting and recovery.

The main class will be a modification of the class `Compiler` used to test the scanner in assignment 1. The main method will call the `ClassDecl parse` routine with stop set including `EOT` to parse the contents of the input file, which must be a class declaration. If successful, there should be no additional tokens in the input (other than the `EOT`). If there is a command line argument, the program should interpret it as the name of the file to be compiled without an extension (i.e. no `.lng`). If there is no command line argument or more than one argument, the program may proceed in any way determined by the programmer, including terminating with an error.

The source file should have the name `xxx.lng` where `xxx` is the file name specified as the command line argument. The compilation process should produce two text files in the same directory as the source file: `xxx.lis` – a listing file with error messages, and `xxx.spm` – a file for generated code. For the purposes of testing and debugging the parser, the code file is used to list the non-terminals (rules) recognized during the parsing process.

For debugging and marking purposes, whenever a syntax error is emitted, an additional message should be emitted to the listing file indicating the point at which the parse continues after the error. This code could be removed in a later phase of the compiler.

Develop a set of test files to test the parser. These should include both valid and invalid input. It is better to have a larger number of small tests than a smaller number of large tests. In each case, it should be easy to determine if the parser passed the test (comments indicating expected output are a good idea).

Run your program against each of your test files and include in your submission:

- source listing of all classes in your program (in alphabetic order by class name within package)
- listing of each source (`.lng`) test file
- listing of each generated listing (`.lis`) file
- listing of each code (non-terminals) (`.spm`) file

Submission

Your submission will be in two parts: electronic and paper. The paper submission should include the listings above and should have a completed coversheet.

In addition to your paper submission, you must make an electronic submission of the assignment. Regardless of how you developed the project, create a directory on sandcastle containing the source (.java) and code (.class) files for your project. In this directory, at the sandcastle prompt, type:

```
submit4p75
```

and, when prompted, enter the appropriate information using 2 as the assignment number. The submission program will copy the entire working directory, and all subdirectories, to the marking directory, so you should ensure that the content of the directory is what you desire to submit. You could create a special directory for submission and only include what is needed, however, make sure the project will run from the command line in that directory before submitting it.

Note: Since the programs will be run versus a test suite, adherence to the program specification and submission instructions is of paramount importance. Submissions in violation of the guidelines may be subject to a penalty.

As usual, the complete paper submission should be placed in an envelope to which a completed coversheet (<http://www.cosc.brocku.ca/coverpage>) is attached. The submission should be made to the **instructor's mailbox in the Department Office** in accordance with the assignment guidelines and not in violation of the regulations on plagiarism (<http://www.cosc.brocku.ca/about/policies/plagiarism>). **Note:** Assignments not including a coversheet will **not** be marked.