

COSC 4P75 Assignment 3

Fall 2011/12

Due: Dec. 5 @ 12:00 noon.

Semantic Analyzer

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

The semantic analyzer is intended for a single-pass compiler and is developed by modification of the classes written for the parser in assignment 2 using the lexical analyzer from assignment 1 as an input filter. 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 for the parser in assignment 2. The main method will call the `ClassDecl` parse routine as before after reading or initializing the global scope. The individual parse routines will be augmented with code for semantic analysis. If there is a command line argument, the program should interpret it as the file name of the file to be compiled, without extension. 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. If there are no errors, the global symbol table should be written (in binary form) to a file `Classes.syt` also in the same directory as the source file(s). For the purposes of testing and debugging the semantic analyzer, the code file should be used to dump the contents of the symbol table (i.e. globals) at the end of the compilation. Each `Declarations` class should have an appropriate `toString` method to allow the dump. The extra message to the listing file indicating the recovery point for syntactic error recovery and the message to the code file indicating which syntactic unit was parsed that were introduced in assignment 2 should be removed in this phase.

Develop a set of test files to test the semantic analyzer. This 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 semantic analyzer 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 (symbol table dump) (`.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 3 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.