Programming Assignment 3

If you have not already set up the VirtualBox VM for a previous assignment, you should do so. Please see the page **Getting started with the VM.PDF** for instructions.

Next, download the **PA3.PDF**. In addition to the assignment requirements, It has information about how to get the starter code up and running.

Please note that this assignment is significantly more algorithmically complex and requires more code than the previous two.

IMPORTANT NOTES:

- **The submission instructions have now been added at the end of this page.**
- **The assignment numbers are offset by 1. Even though this is assignment 3, the directory that you will get the starter code from is named "PA4" for the C++ version and "PA4J" for the Java version.**

Here are some more useful resources:

- **The Cool Reference Manual.PDF**
- **A Tour of the Cool Support Code.PDF** In particular, look at section 6 "Abstract Syntax Trees".
- Some additional **Other Project Resources.PDF** including miscellaneous documentation.

As with assignments 1 and 2, the examples are in the directory /usr/class/cs143/examples. Please copy these into your project directory if you wish to use them.

The correct version of spim is at /usr/class/cs143/bin/spim.

Once your semantic analyzer works, you should be able to compile the examples using your semantic analyzer and run them. As an example, the following should work (some output omitted):

```
$ make semant
...
$ cp /usr/class/cs143/examples/hello_world.cl .
```

```
$ ./mycoolc hello_world.cl
$ /usr/class/cs143/bin/spim hello_world.s
SPIM Version 6.5 of January 4, 2003
Copyright 1990-2003 by James R. Larus (larus@cs.wisc.edu).
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/class/cs143/lib/trap.handler
Hello, World.
COOL program successfully executed
Stats -- #instructions : 154
#reads : 27 #writes 22 #branches 28 #other 77
```

Unfortunately, in the process of packaging the starter code for the public class, we introduced a bug that causes the reference parser to report incorrect line numbers. Since the semantic analyzer relies on the reference parser to supply it with line numbers, the grading script will not check the line numbers in the error messages produced by your semantic analyzer.

How to Submit

1. Download the grading script from here and put it in the directory in which you are doing the assignment (where the semant.cc or Semant.java file is). The easiest way to do so is to go to your assignment directory, and run in the VM :

   ```
   wget
   http://spark-university.s3.amazonaws.com/stanford-compi
   lers/scripts/pa3-grading.pl
   ```

   or

   ```
   wget
   https://lagunita.stanford.edu/assets/courseware/v1/9334
   cd3f5353fabc5cc62247b2ae0a1d/c4x/Engineering/Compilers/
   asset/pa3-grading.pl
   ```

   This will save the script (pa3-grading.pl) in your assignment directory.

2. Run the script by typing

   ```
   perl pa3-grading.pl
   ```

Note that you can also make the script executable by running `chmod a+x pa3-grading.pl` first, and then running it directly as `./pa3-grading.pl`

3. The script will give you a grade at the end, as well as a submission code. If you want to figure out why your semantic analyzer is failing certain tests, the tests will be put in the ./grading subdirectory. The output from your code will be in the ./grading/test-output directory.

4. Once you are satisfied with your grade, click on the arrow above or beneath to go to the "Programming Assignment 3 Submission" quiz. You can use this link to go directly to the quiz. Copy-and-paste the code from the script (to copy from the terminal in VirtualBox, use ctrl+shift+c) into the "Submission code:" box. Once you submit the quiz, your score should appear for the quiz. You can also resubmit the quiz if you wish to update your grade.