

CSE 5343, Programming Project 0

Due Tuesday, January 16, 11:59 pm (0 points)

The goal of this project is to set up your execution environment for the real programming projects. *Even if you have used similar setups for CSE 3341 or CSE 6341, please create the environment for CSE 5343 from scratch.*

Please do the following:

1) Ensure that you can log into UNIX server `stdlinux.coeit.osu.edu`. For simple SSH access, use PuTTY or a similar client. Alternatively, use FastX. More details under “Resources” on the CSE 6341 web page: <https://web.cse.ohio-state.edu/~rountev.1/6341/resources.html>

2) From a terminal window on `stdlinux`, run

```
subscribe
```

Check your software subscriptions. If you are not already subscribed to JDK-CURRENT, subscribe to it. Log out, then log in again, and do

```
java -version
```

You should see something like

```
openjdk version "14.0.1" 2020-04-14
OpenJDK Runtime Environment (build 14.0.1+7)
OpenJDK 64-Bit Server VM (build 14.0.1+7, mixed mode, sharing)
```

3) Create a directory for the project and download a skeleton implementation. The examples are for username `buckeye.8`; obviously, replace with your own username.

Let’s say you have created `/home/buckeye.8/5343` for this project.

```
cd /home/buckeye.8/5343
wget web.cse.ohio-state.edu/~rountev.1/5343/project/proj.tar.gz
tar -xvzf proj.tar.gz
cd proj
```

4) Set up two environment variables. How you do this depends on what Unix shell you are using. If you are not sure, do

```
echo $SHELL
```

Set up the following variables:

```
JFLEX_DIR should be set to /home/buckeye.8/5343/proj/jflex-1.7.0
CUP_DIR should be set to /home/buckeye.8/5343/proj/cup
```

For example, in `bash` you can use `export VARNAME="my value"`

Make these permanent by adding them to your `.bashrc` or similar file; if you are not sure what this means, see “Resources” on the web page for more details on environment variables.

5) Log out and then log in again. Print the values of environment variables with

```
printenv
```

You should see the correct values for `JFLEX_DIR` and `CUP_DIR`

6) Go to directory `proj/p1` and do `make`. The result is a parser/scanner combo for a toy language we will call simpleC. The language is a subset of the C language.

7) Execute the parser on a toy simpleC program, which is in file `t1.c`
`./simplec t1.c`

This should produce pretty printing of the abstract syntax tree (AST) of the input program.

8) Create another simpleC program by adding the following text to file `proj/p1/t2.c`

```
double g() { int x; int y;  
    x = 5+ 3; y = x; }
```

Run `./simplec t2.c` and record the output.

9) Create the following program in text file `proj/p1/t3.c`

```
double h() { int x+5 = 3; }
```

Run `./simplec t3.c` and record the output.

10) In Carmen, submit these two outputs in a single text file