Project 1: Lexical Analysis

CSC 4351, Spring 2014

Due: 5 February 2014

Use JLex to implement a lexical analyzer for the Tiger language. You can find the description for this project on pages 34 and 35 of the first edition of the textbook. The specification of the lexical tokens is in the *Tiger Language Reference Manual* in Appendix A.

Environment and Support Files

Define the following environment variables (in bash syntax) in your `.profile` file:

```
export CS4351=/classes/cs4351/cs4351_bau/pub
export PROG=chap2
export TIGER=${CS4351}/tiger
export CLASSPATH=.:.:.:${CS4351}/classes/${PROG}:${CS4351}/classes
```

The directory `${CS4351}` contains publicly available files for this class. The variable `PROG` names the book chapter containing the project description and is used for setting the `CLASSPATH`. The environment variable `TIGER` points to a directory containing code skeletons for the projects.

The `CLASSPATH` contains four directories. The current and the parent directories are used for finding the compiled files of your own compiler when you are in your working directory. The directory `${CS4351}/classes/${PROG}` contains the class files for a solution to the project. If you want to run the solution, you need to go outside your working directory. If you are in your working directory, your own class files will be found ahead of the ones I provided. The directory `${CS4351}/classes` contains the class files for JLex and Java CUP.

The file `java_cup/runtime/Symbol.java` the textbook refers to is not in `${TIGER}/chap2` but in `${CS4351}/classes`.

Compilation

To get started, copy the files provided in `${TIGER}/chap2`:

```
cp -rp ${TIGER}/chap2 prog1
cd prog1
```

You can then use `make` to compile from the sources. We use `make` only for compiling non-Java sources (i.e., `Tiger.lex` for the scanner and later `Grm.cup` for the parser). For recompiling individual Java files, you would run `javac` manually, as in

```
javac -g package/file.java
```

The option `-g` causes `javac` to include debug information in the class file so you can debug your program using `jdb`.

For running your scanner on an input file `test.tig`, execute

```
java Parse.Main test.tig
```

in your working directory.

Submission

For this project, you only need to modify file `Parse/Tiger.lex`. To save space in the submit directory, please, don't submit any class files. E.g.,

```
cd prog1
make clean
~cs4351_bau/bin/p_copy 1
```

Please provide a README file with your submission. In the README file, provide any information that will help the grader to give you partial credit. I.e., state what works, what doesn't work, and what implementation decisions you made. In particular, mention how you handle comments, strings, and control characters. A paragraph or two should be enough. Also, don't forget to list the team members.