Assignment # 2
COP 3402 Fall 2011
Lexical Specification of KOOL Language (FLEX this time)

You are to start with my FLEX input and fix up its several problems (not handling underscore, not limiting ident values to 8 characters, not looking that integer constants terminate on a white space or special character, not doing the same for identifiers, not reporting the decimal value, rather than the corresponding string for an integer constant). Most of the rest is from assignment # 1. Notice, I added **private** as a keyword.

KOOL has a relatively small set of keywords. At present (and life can change over the semester), these reserved words are:
void int bool string null true false class new this extends
for do while until if then else break return
print println private read readln

Special characters in **KOOL** are:
```
+   -   *   /   %   =   +=   -=   *=   /=   %=   <   <=   >   >=   ==   #   ||
&&   !   ;   ,   .   [   ]   (   )   {   }
```
Notice that neither " nor \ are special characters as they can only appear as part of a string. _ is also not a special character as it is treated as a null character in an identifier.

Anything that is a blank or that collates to being less than a blank is whitespace. These characters separate tokens.

An integer constant can either be specified in decimal (base 10) or binary (base 2). A decimal integer is a sequence of decimal digits. A binary integer must begin with 0B or 0b and is followed by a sequence of binary digits. All integer constants must end with a whitespace or special character.

An identifier is a sequence of letters, digits, and underscores, starting with a letter. This is case insensitive. The underscores are for readability and should be removed. Identifiers that match in their first 8 letters are considered the same, so you will want to truncate them to this maximum length. Keywords and identifiers must be terminated by whitespace or a special character. Thus, whiledo is an identifier rather than two keywords. MyData(123#x) might make no sense syntactically, but is viewed lexically as IDENT(mydata), LPAREN, INTEGER(123), NOTEQUAL, IDENT(x), RPAREN.

A string constant is a sequence of characters enclosed in double quotes. Strings can contain any character except a newline or double quote, unless the double quote is preceded by an escape character (\). The escape character just says that the next character is part of the string; the escape itself is not. Thus, to include a backslash, you need to escape it. A string cannot span multiple lines nor can it exceed 80 characters, including the quotes and escape characters – this means strings are limited to 78 characters each.

An end-of-line terminated comment is started by // and extends to the end of the line. Any symbol is allowed after the // with the end-of line signifying its end. C-style comments start with /* and end with the first subsequent */. Any symbol is allowed in a C-style comment except the sequence */ which ends the current comment. C-style comments can span multiple lines, but also can end in the middle of a line.

Your lexical analyzer must report on each token, giving the token type and, where appropriate, the token value. You do NOT need to echo the input.

Turn in a FLEX input file (typically a .l file). Name your file with your First Initial followed by your Last Name followed by "Asn2" and the FLEX extension. E.g., Steven would be SZittrowerAsn2.l.

Due Dates:

Due on Monday, October 3 before the end of day (11:59PM)

Turn in must be done via Webcourses. Only the source is to be turned in..

PIPELINE POINTS (20 POINTS)

- The student's submitted file will be converted to a C file using Flex. This C file will then be compiled into an executable. These steps should not cause any errors. Since this assignment is so small, no partial credit is granted. However, I will still try to fix small errors if I can see your intent.
- Flex generates C file from student's lex file – 10 points
- C file can be compiled into executable – 10 points

EXECUTION POINTS (50 POINTS)

FLEX IMPLEMENTATION DETAILS (30 POINTS)

We are looking for correct functionality (at least conceptually).

NOTES:

If the assignment was submitted up to 24 hours late, there is a 10% deduction from the final grade; submitting between 24 – 48 hours late results in a 20% deduction.