CPSCI 323
Programming Languages and Translation
Fall 2014

Program #1, due Wednesday. 9/10

Using C++, implement a lexical analyzer (Lexer) for a language with the specifications given below.

Include a main program that will read in the name of the file that is to be analyzed. The main program should repeatedly ask the Lexer to provide the next lexeme/token pair from the input file. The main program will then write out the lexeme and the token before calling the Lexer again. If an invalid input is found, write out the lexeme and "invalid" and end the program.
This process will continue until an invalid input is found or the end of the input file is reached.

Language Specifications:

Identifier: a sequence of letters, digits, and underscore“_”
The first character must be a letter
The last character may not be an underscore

Integer: an unsigned series of digits (ex: 123)

Real: an unsigned series(1 or more) of digits, followed by “.”, followed by another unsigned series(1 or more) of digits (ex: 12.34)

Keywords: int, real, bool, program, begin, end, function, read, write, if, else, elseif, while, do, until, true, false

Math Operators: + − * /

Relational Operators: > < >= <= = **<>**

General Operators: <− ; : , ( ) . !

Comment Markers: //

Example: Input file:

```
while ( high_1 > 127 )        //  first comment
   begin
     n <− 15.24 ;             //   second comment
   end
```

Output:

```
while        keyword
(            operator
high_1       identifier
>            rel. op
127          integer
)            operator
begin        keyword
n            identifier
<−           operator
15.24        real
;            operator
end          keyword
```