CS 280
Fall 2022

Recitation Assignment 5
October 13, 2022

Due Date: Monday October 17, 2022, 23:59
Total Points: 6

You are given a copy of "`lex.h`" from Programming Assignment 1, and a file called "`tokensListing.cpp`" as a driver program.

DO NOT CHANGE neither "lex.h" nor "`tokensListing.cpp`".

Your implementation should include the following in another file, called "`RA5.cpp`":

- The function

```
LexItem id_or_kw(const string& lexeme, int linenum);
```

`id_or_kw()` function accepts a reference to a string of a lexeme and a line number and returns a LexItem object. It searches for the lexeme in a directory that maps a string value of a keyword to its corresponding Token value, and it returns a LexItem object containing the keyword Token if it is found and not equal to a TRUE or FALSE, or the identifier token IDENT if not. However, if the string value of a keyword corresponds to either the TRUE or FALSE tokens, the function should return a LexItem object containing the BCONST token instead.

- The overloaded operator function operator<< for `LexItem`.

```
ostream& operator<<(ostream& out, const LexItem& tok);
```

The `operator<<()` function accepts a reference to an ostream object and a reference to a LexItem object, and returns a reference to the ostream object. The operator<< function prints out a LexItem object information according to the Token value using the following formats:

Token	Format
IDENT	IDENT: <lexeme> at Line <linenumber>
Keyword	KEYWORD: <token> at Line <linenumber>
ICONST, RCONST, BCONST	<TOKEN>: (<lexeme>) at Line <linenumber>
SCONST	SCONST: "<lexeme>" at Line <linenumber>
Operators	<TOKEN>: '<lexeme>' at Line <linenumber>
ERR	ERROR: "<lexeme>" at Line <linenumber>

Note that the implementation of operator<< function in RA5 differs from its implementation in PA1. See the examples below for the output format.

Use the given driver program in "`tokensListing.cpp`" for testing your implementations. The driver program accepts two command line arguments, "-othertok" and "-idsonly".

The "-alltok" flag is used to test your implementation of the overloaded operator<< function. While, the "-idsonly" is used to test your implementation of `id_or_kw ()` function. See the details of the outputs for the examples in the slides.

Submission Guidelines

1.1. Please upload your RA5.cpp file to Vocareum. The "`lex.h`" header file and "`tokensListing.cpp`" driver program will be propagated to your Work Directory.

1.2. Submissions after the due date are accepted with a fixed penalty of 25% from the student's score. No submission is accepted after Wednesday 11:59 pm, October 19, 2022.

Grading Table

Item	Points
Compiles Successfully	1
Implementation of `id_or_kw` Function	2
Implementation of operator<< Function	3
Total	6