

**CS 280**  
**Spring 2023**

**Recitation Assignment 5**  
**February 23, 2023**

**Due Date: Monday February 27, 2023, 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. Otherwise, it returns a LexItem object containing a token for one of the possible types of identifiers (i.e., IDENT, SIDENT, or NIDENT).

- 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	IDENTIFIER: <lexeme> at Line <linenumber>
NIDENT/SIDENT	<Token>: “<lexeme>” at Line <linenumber>
Keyword token	<Token>: “<lexeme>” at Line <linenumber>
ICONST/ RCONST	<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 in the class slides for the output format.**

Use the given driver program in “tokensListing.cpp” for testing your implementations. The driver program accepts a file name for a test case in each run, and there are 5 test cases used. 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, March 1<sup>st</sup>, 2023.

### **Grading Table**

Item	Points
Compiles Successfully	1
Case 1	1
Case 2	1
Case 3	1
Case 4	1
Case 5	1
Total	6