CS 3342 Lab #2 - Lexical Analysis in Python and C

Due Date: February 20, 2022 (Sunday) - 11:59pm

LAB 2A – Lexical Analysis in Python (50 points)

(https://www.w3schools.com/python/python_intro.asp - Python for beginners)

1) Copy a Python program called 'python_lexer_student.py' and an input file called 'lab2_test.c' from the lab2 folder on Canvas.

The Python program is a lexer which will take an input character stream and convert it into tokens. Read and try to understand the program. Run the program using Visual Studio Code or an IDE/IDLE you prefer. The program will read the input file and print the following output:

```
IDENTIFIER(int) at 0
IDENTIFIER(main) at 4
LP(() at 8
RP()) at 9
invalid token on this line at  11 : int main() {
IDENTIFIER(int) at 3
IDENTIFIER(x) at 7
invalid token on this line at  8 :    int x,y;
IDENTIFIER(float) at 3
IDENTIFIER(test_z) at 9
EQUALS(=) at 16
NUMBER(100) at 18
invalid token on this line at  21 :    float test_z = 100;
IDENTIFIER(int) at 3
IDENTIFIER(c_id) at 7
EQUALS(=) at 12
NUMBER(3342) at 14
invalid token on this line at  18 :    int c_id = 3342;
IDENTIFIER(x) at 3
EQUALS(=) at 5
NUMBER(4) at 7
PLUS(+) at 9
NUMBER(5) at 11
invalid token on this line at  12 :    x = 4 + 5;
IDENTIFIER(y) at 3
EQUALS(=) at 5
NUMBER(6) at 7
MULTIPLY(*) at 9
NUMBER(7) at 10
invalid token on this line at  12 :    y = 6 *7 ;
IDENTIFIER(return) at 3
NUMBER(0) at 10
invalid token on this line at  11 :    return 0;
invalid token on this line at  1 :  }
```

Your task is to modify the Python program to fix the invalid token errors and to print the following output with the same input file.

Lexeme	Token
int	(KEYWORD)
main	(IDENTIFIER)
(	(LPAREN)
)	(RPAREN)
{	(LBRACE)
int	(KEYWORD)
x	(IDENTIFIER)
,	(COMMA)
y	(IDENTIFIER)
;	(SEMICOLON)
float	(KEYWORD)
test_z	(IDENTIFIER)
=	(EQUALS)
100	(NUMBER)
;	(SEMICOLON)
int	(KEYWORD)
c_id	(IDENTIFIER)
=	(EQUALS)
3342	(NUMBER)
;	(SEMICOLON)
x	(IDENTIFIER)
=	(EQUALS)
4	(NUMBER)
+	(PLUS)
5	(NUMBER)
;	(SEMICOLON)
y	(IDENTIFIER)
=	(EQUALS)
6	(NUMBER)
*	(MULTIPLY)
7	(NUMBER)
;	(SEMICOLON)
return	(KEYWORD)
0	(NUMBER)
;	(SEMICOLON)
}	(RBRACE)

Take a screen shot of your program output and put in a word document. Zip your source program. You can put your outputs from Lab2A and Lab2B in one word doc. Submit the word doc and zip files separately to Canvas.

Lab 2B - Lexical Analysis in C (50 points)

Software/tools needed:
1. Visual Studio Code or an IDE for C you prefer:
2. gcc (for c/c++)
   a. **For Mac: gcc should be already there. If not, try the following:**
      a. https://www.youtube.com/watch?v=0z-fCNNqfEg (installing gcc in mac)

         i. Download Visual Studio Code for Mac OS X.
         ii. Double-click on VSCode-osx. zip to expand the contents.
         iii. Drag Visual Studio Code app to the Applications folder, making it available in the Launch pad.
         iv. Add VS Code to your Dock by right-clicking on the icon and choosing Options, Keep in Dock.
   b. **For Windows:**
      a. https://www.youtube.com/watch?v=MlIzFUI1QGA (install VS Code)
      b. Install C/GCC Compiler for Windows - JournalDev (install C/GCC compiler)
      c. https://www.youtube.com/watch?v=guM4XS43m4I (install c/c++ compiler – old verion)
      d. https://www.youtube.com/watch?v=77v-Poud io (run c/c++ program – example)

(Note: In windows: if get the following message or the windows defender does not allow you to run this program in VS code, you need to click in the defender option to allow it to run.
"Program 'xxxxx.exe' failed to run: Operation did not complete successfully because the file contains a virus or potentially unwanted software At line:1 char:90"**)**

Steps:
1) Copy the program called 'c_lexer_student.c' from the lab2 folder on Canvas. This C program is a lexer which will take an input character stream and convert it into tokens.

Read and try to understand the program. Run the program in Visual Studio Code or an IDE you prefer. The program will read the same input file in Lab2A (lab2_test.c). This program does not skip comments and will print the following output:

```
/ is an operator
/ is an operator
Name is an identifier
: is an identifier
Your is an identifier
Name is an identifier
/ is an operator
/ is an operator
Class: is an identifier
CS is an identifier
3342 is a number
Programming is an identifier
Languages is an identifier
```

/ is an operator
/ is an operator
Lab is an identifier
1B is a number
Sample is an identifier
Program is an identifier
/ is an operator
/ is an operator
Due is an identifier
Date: is an identifier
September is an identifier
13 is a number
2020 is a number
Sunday is an identifier
int is a keyword
main is an identifier
int is a keyword
x is an identifier
y is an identifier
float is a keyword
test_z is an identifier
= is an operator
100 is a number
int is a keyword
course_num is an identifier
= is an operator
3342 is a number
x is an identifier
= is an operator
4 is a number
+ is an operator
5 is a number
y is an identifier
= is an operator
6 is a number
* is an operator
7 is a number
return is a keyword
0 is a number

Your task is to modify the program to skip comments and print the following output:

Lemexe	**Token**
int	keyword
main	keyword
(	lparen
)	rparen
{	lbrace
int	keyword
x	identifier
,	comma
y	identifier
;	semicolon
float	keyword
test_z	identifier
=	operator

100	number
;	semicolon
int	keyword
c_id	identifier
=	operator
3342	number
;	semicolon
x	identifier
=	operator
4	number
+	operator
5	number
;	semicolon
y	identifier
=	operator
6	number
*	operator
7	number
;	semicolon
return	keyword
0	number
;	semicolon
}	rbrace

Notice that the output is slightly different from the one in Lab2A. For example,

1. lowercase versus uppercase
2. "=" is categorized as "operator' here and it is "EQUALS" in Lab2A.

Take a screen shot of your program output and put in a word document. Zip your source program. You can put your outputs from Lab2A and Lab2B in one word doc. Submit the word doc and zip files separately to Canvas.

The following is the sample input file:

```
//Name: Your Name
//Class: CS 3342 Programming Languages
//Lab 1B Sample Program
//Due Date:  September 13, 2020 (Sunday)

int main() {
   int x, y;
   float test_z = 100;
   int course_num = 3342;

   x = 4 + 5;
   y = 6 * 7;
   return 0;
 }
```