GABINETE, KEITH GINOEL S. **CMSC 129 - CD-1L**

Exercise 5: yacc

Instructions:

1. Create a lex and yacc/bison program of your favorite Programming language that will check if you have the correct syntax and semantics for basic math operations and printing strings
    - Addition
    - Subtraction
    - Multiplication
    - Division
    - Printing “Hello World”
2. Submit instructions on how to compile and run your program
3. Submit screenshots of your sample output

NOTE:

For the instructions on how to compile and run the program, refer to the 'README.md' file included in the zip file.

SCREENSHOTS

OPERATIONS:

```
shio@X509FB:~/Desktop/repos/principles-of-compiler-design/05 - YACC Introduction/GABINETEKG_exer5$ make
yacc -d GABINETEKG_exer5.y
lex GABINETEKG_exer5.l
cc y.tab.c lex.yy.c -o GABINETEKG_exer5.exe
./GABINETEKG_exer5.exe
═══════════════════════════════════════════════════════════════════════
This is a simple program that checks the syntax and semantics of basic
mathematical operations and string printing using lex and yacc, following
Python rules.

Basic operations supported include addition, subtraction, multiplication,
division, and simple printing of strings.

Use print() to print a string.
Strings must be enclosed in either double or single quotes.
If a string is enclosed in double quotes, it can contain single quotes.
If a string is enclosed in single quotes, it can contain double quotes.
Only one string parameter is supported at a time.

To terminate the program, just type 'exit()'.
═══════════════════════════════════════════════════════════════════════
>>> 8 + 4 + 2
14
>>> 8 - 4 - 2
2
>>> 8 * 4 * 2
64
>>> 8 / 4 / 2
1
>>> 1 + 2 * 2 / 4 * 8 + 1
10
>>> print("hello world")
hello world
>>> print('world hello')
world hello
```

ERRORS:

```
>>> 8 -* 2
syntax error
>>> print('This is invalid")
0
syntax error
>>> print('
0
syntax error
>>> print("This is invalid")in
syntax error
>>> print("This is " invalid")
0
syntax error
```

Using single-letter variables:

```
>>> a = 2
>>> b = 3
>>> c = 4
>>> a + b
5
>>> a - c
-2
>>> b * c
12
>>> c / a
2
>>> c * 8 + b - 4 / a
33
```

Exit:

```
>>> exit()
Exiting program... Goodbye!
```