UNIVERSITY OF NAIROBI

FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTING AND INFORMATICS

CSC 326: COMPILER CONSTRUCTION

SCANNER FOR OUR MINI LANGUAGE USING PLY TOOL

Done by Group 17:

P15/137032/2019	Hamud Abdulrahman Abeid
P15/1894/2020	Emmanuel Mbithi Muthiani
P15/1915/2020	Mugo Staicy Nelima Muthoni

SOURCE CODE FOR THE SCANNER IN PYTHON

```
import ply.lex as lex


keywords = {

   'if'     : 'IF',

   'else'   : 'ELSE',

   'while'  : 'WHILE',

   'print'  : 'PRINT'

}


# Define tokens

tokens = [

    'ASSIGN',

    'PLUS',

    'MINUS',

    'MULT',

    'DIV',

    'LPAREN',

    'RPAREN',

    'LBRACE',

    'RBRACE',

    'ID',

    'INT',

    'FLOAT',

    'EOL',

    'LT',

    'GT',

    'LE',
```

```
    'GE',
    'EQ',
    'NE'
] + list(keywords.values())

# Define regular expression rules for tokens
t_ASSIGN = r'='
t_PLUS = r'\+'
t_MINUS = r'-'
t_MULT = r'\*'
t_DIV = r'/'
t_LPAREN = r'\('
t_RPAREN = r'\)'
t_LBRACE = r'{'
t_RBRACE = r'}'
t_FLOAT =  r'\d+\.\d+'
t_INT = r'\d+'
t_EOL = r';'
t_LT = r'<'
t_GT = r'>'
t_LE = r'<='
t_GE = r'>='
t_EQ = r'=='
t_NE = r'!='

# Define rule to check for keywords
def t_ID(t):
    r'[a-zA-Z_][a-zA-Z0-9_]*'
```

```
    t.type = keywords.get(t.value, 'ID')
    return t

# Define ignored characters (whitespace)
t_ignore = ' \t'

# Define new line tracking rule
def t_newline(t):
    r'\n+'
    t.lexer.lineno += len(t.value)



# Define error handling rule
def t_error(t):
    print(f"SyntaxError: Invalid token '{t.value[0]}' at line {t.lineno}")
    t.lexer.skip(1)

# Build lexer
lexer = lex.lex()

# Test lexer with sample input
data = '''x = 42.4;
y = 78;
a = 0.98\n
 while: (x > 0) {
    print(x)[];
    x = x - 1;
}
```

```
'''


# Run the lexer

lexer.input(data)


# Tokenize

for token in lexer:

    print(token)
```

SCREENSHOT SHOWING TOKENS AND SCANNERS RUN-TIME