CC 411 Compiler Construction

(Deadline: November 07, 2024 by 1400 hrs)

Assignment No: 1 Building a Lexer for the MiniLang Language

You are required to design and implement a Lexical Analyzer (Lexer) in a programming language of your choice like Python, Java or C++ to process and tokenize code written in a small hypothetical programming language called **MiniLang**. This Lexer will be the first step in building a full compiler for **MiniLang**.

MiniLang Features

MiniLang is a simplified programming language designed for this a semester level course on compilers. The language includes the following types of tokens:

Keywords: `if`, `else`, `while`, `print`, `int`, `float`

Operators: `+`, `-`, `*`, `/`, `=`, `==`, `!=`, `<`, `>`, `<=`, `>=`

Delimiters: `(`, `)`, `{`, `}`, `;`, `,`

Identifiers: Variable names that start with a letter, followed by letters or digits (e.g., `var1`, `total`)

Literals: Integer literals (e.g., `123`), floating-point literals (e.g., `45.67`), and string literals enclosed in double quotes (e.g., `"Hello World"`)

Comments: Single-line comments starting with `//`

Requirements for this Assignment

1. **Token Classification**:

Your lexer should read an input **MiniLang** source code file, identify each token, and classify it into its type (e.g., keyword, operator, identifier, literal).

2. **Output**:

The lexer should output a list of tokens in the format (`token_type`, lexeme), where:

`token_type` is the type of the token (e.g., `KEYWORD`, `OPERATOR`, `IDENTIFIER`, `LITERAL`, etc.).

- `lexeme` is the exact string or value of the token.

For example, if the input is:

```
int x = 10;
 if (x > 5) {
     print("Hello World");
 }
```

The output should look like:

(KEYWORD, "int")

(IDENTIFIER, "x")

(OPERATOR, "=")

(LITERAL, "10")

(DELIMITER, ";")

(KEYWORD, "if")

3. **Error Handling:**

You should include basic error handling for invalid tokens or unrecognized characters and display an error message with the **line number**.

4. **Comments:**

Skip over single-line comments and do not include them in the output.

5. **Testing:**

Create test cases that include examples of valid and invalid **MiniLang** code and use them to test the lexer.

Bonus Task (Optional for more enthusiastic compiler designers 😊):

Extend the lexer to handle multi-line comments, string escape characters, or additional operators.

The assignment will be graded on the basis of a viva in the class on November 07, 2024, therefore, it is advised to do it yourself by applying your skills learnt so far in your BS and making use of good programming practices and code conventions for Object Oriented Programming. Use of ChatGPT is a big NO.

Good Luck 😊