**DESCRIBING OUTPUT OF A SCANNER**

In a compiler, there are several phases that a source program undergoes before a target program is output. The first phase entails lexical analysis. Lexical analysis is performed by a lexical analyzer or a scanner.

The scanner normally reads the source program and converts the characters to meaningful lexemes which are presented in form of tokens. The different types of tokens include:

1. Constants
2. Identifiers
3. Operators
4. Separators
5. Reserved words.

To further describe this, we will use the following input: **a+10=j;**

**Constants**

This includes any possible combination of numbers from 0-9. This can also be represented as [0-9]+. For example, 10,4027…etc. In our example code then, the constant will be: 10

**Identifiers**

An identifier includes all the possible combination of uppercase and lowercase of the 26 alphabets. This is normally represented as: [A-Za-z]+. Therefore, an identifier can be a string of letters like juju or a single letter j. In our example, the identifiers are: a and j.

**Operators**

These are mathematical symbols that are used for addition, subtraction, division and multiplication, equal to, less than or equal to, greater than and equal to or greater than. They can be represented as followers in a lexical analyzer: "\\\*|\\+|\\-|\\=". More operators can be included. From our example, the operators are + and =.

**Separators**

This include symbols like comma, semi-colon. They are used to show when a line of code ends or mark a different part. For example, in a+10=j; the semicolon shows that the end of the line of code is after j.

**Reserved words**

These are special words that have some functions assigned to them in a programming language. There are several reserved words in programming. Some of them are: BREAK, CASE, FOR, DOUBLE, AUTO, FLOATE, VOID, STATIC, among others. We do not have a reserved word in our example but that is it.

I will show my lab 1 output as another example:

