|  |  |
| --- | --- |
| **DESCRIBING SCANNER OUTPUT (LEXICAL ANALYZER)**  **MIRIAM MMBOGA 101534 ICS4A**  **13/11/2020** | Compiler construction tools - GeeksforGeeks |

**1.1 Introduction**

A lexical analyzer which is a scanner is the accepts source code and generates compiler specific tokens from it. The source code (in a specific programming language) is taken as the input for the scanner and the ‘output’ are the tokens which are passed to a parser. In the scanner, generated tokens follow rules of mathematical expressions. This means that a scanner can recognize operations like +,/,-,%,= and \*. Additionally, a scanner can recognize language constructs like alphabets which are found in function calls, declarations (main) and data types e.g. (int).

**1.2 Lab work**

In this lab, a text file named ‘mysamplecode.txt’ contains the code fragment to be analyzed. This code applies a formula to calculate net pay in the main function declaration part. The code is in C++.

void main()

{

int gross, net ,tax;

net = gross - tax;

}

**1.3 Explanation**

As mentioned above, a scanner can identify alphabets, numbers and mathematical operations. From the code, these are identified as:

* To scan for keywords, the function isKeyword () is used to check and store the keywords identified e.g void.
* To scan for alphabets and numbers, the function isalnum() is used to check and store 0-9, a-z and A-Z.
* To scan for mathematical operations, all the operators are declared (+-=%/\*) in an array and a loop iterates the code checking for anywhere these symbols appear.