Department of Computer Science and Engineering

Course Code: CSE 420	**Credits: 1.5**
Course Name: Compiler Design	**Semester: Summer 2020**

Lab 01

Introduction

I. Topic Overview:

The lab is designed to introduce the students to the basics steps of a compiler Design. After that the lab will focuses and dive deep into the first compilation step which is Lexical analysis. Basic techniques of coding and required tools will also be shown to students.

II. Lesson Fit:

The lab gives a hand on experience of the knowledge of theory class.

III. Learning Outcome:

After this lecture, the students will be able to:

a. Understand and visualize the Lexical Analysis step.

b. Identifying the "Token" for a particular program.

c. Creating own version of Lexical analyzer.

IV. Anticipated Challenges and Possible Solutions

a. Finding out the identifiers and numerical values from the given code. **Possible Solutions:**

i. Use regular expression.

ii. Use methods of java String class.

V. Acceptance and Evaluation

If a task is a continuing task and one couldn't finish within time limit, he/she will continue from there in the next Lab, or be given as a home work. He/ she have to submit the code and have to face a short viva. A deduction of 30% marks is applicable for late submission. The marks distribution is as follows:

Code: 0%

Viva: 100%

VI. Activity Detail

a.Hour: 2

Discussion: Basic Concepts & Regular Expressions

1. What does a Lexical Analyzer do?
2. How does it Work?
3. Formalizing Token Definition & Recognition

Problem task:

i. Task 1 (Page 3- 4)

Assignment 1: Problem Description

In this assignment, your job is to program a simple lexical analyzer that will build a symbol table from given stream of chars. You will need to read a file named "input.txt" to collect all chars. For simplicity, input file will be a C program without headers and methods. Then you will identify all the numerical values, identifiers, keywords, math operators, logical operators and others

[distinct]. See the example for more details. You can assume that, there will be a space after each keywords. But, removal of space will add bonus point.

Lab 1: Activity List

Task 1: For the following sample program find all the tokens.

Note that a ***lexical Analyzer*** is responsible for

1. Scan Input
2. Remove WS, NL, ...
3. Identify Tokens
4. Create Symbol Table (ST)
5. Insert Tokens into ST
6. Generate Errors
7. Send Tokens to Parser

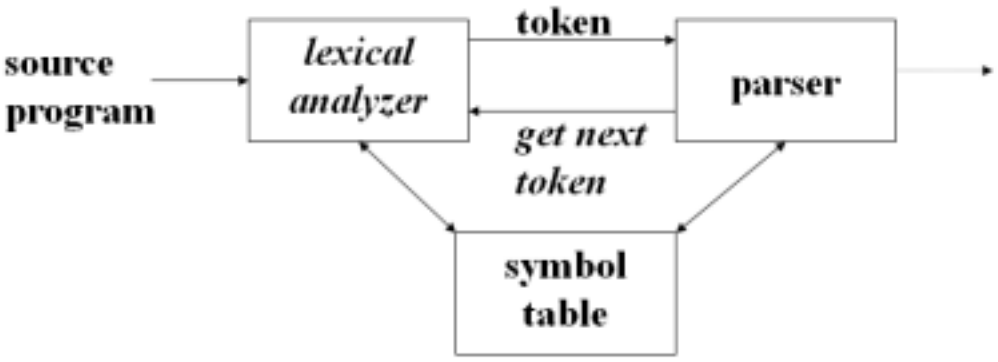

Fig 1: Lexical Analyzer

Input:

```
int a, b, c;
float d, e;
a = b = 5;
c = 6;
if ( a > b)
```

```
{
 c = a - b;
        e = d - 2.0;
}
else
{
 d = e + 6.0;
        b = a + c;
}
```

Output:

Keywords: int, float, if, else

Identifiers: a, b, c, d, e

Math Operators: +, -, =

Logical Operators: >

Numerical Values: 5, 6, 2.0, 6.0

Others: , ; ( ) { }