Department of Computer Engineering

Academic Term: Jan-May 22-23

Class: T.E. (Computer)
Subject Name: System Programming and Compiler Construction
Subject Code: (CPC601)

Assignment No:	**2**
Mapped to CO:	Write a program to implement Lexical analyzer
Date of Submission:	**10-02-2023**
Roll No:	9194
Name of the Student:	Suzan Dsouza

Evaluation:

Sr. No	Rubric	Grade
1	Time Line (2)	
2	Organization (2)	
3	Level of content (4)	
4	Depth and breadth of discussion (2)	

Signature of the Teacher:

Experiment No 2

Aim: Write a program to implement Lexical analyzer

Code:

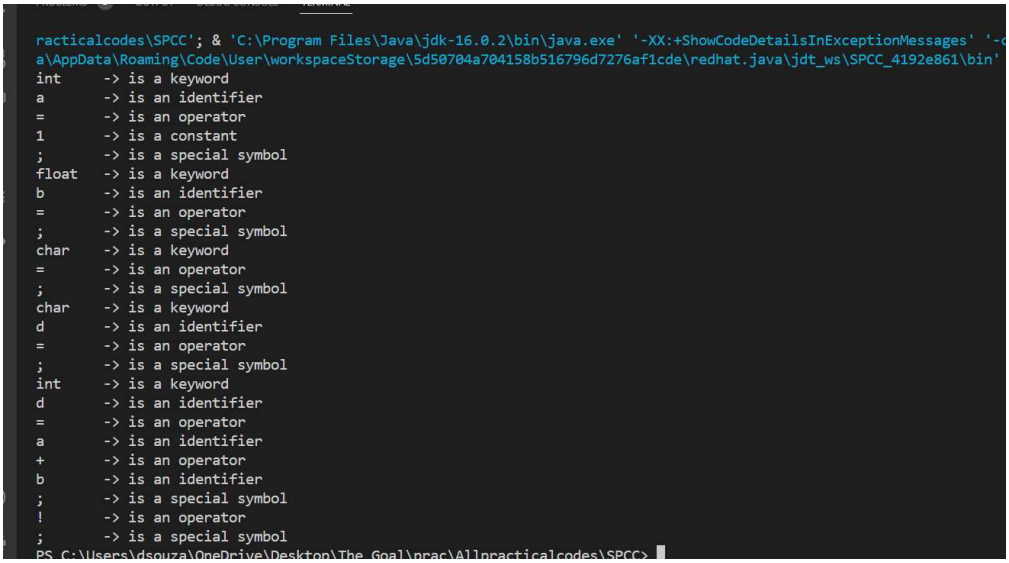
```
import java.io.FileInputStream;
import java.io.InputStream;
import java.util.Scanner;
public class pract2 {
   public static void main(String[] args) throws Exception{
      InputStream inpStream=new FileInputStream("./input.txt");
      Scanner sc=new Scanner(inpStream);
      StringBuilder sb = new StringBuilder();
      while(sc.hasNext()){
            sb.append(" "+sc.next());
      }
      String[] formatted_sb=sb.toString().split(" ");
      for(String s:formatted_sb){
         //System.out.println(s);

         if(s.matches(("[,/!%<>=*+-]"))) {
            System.out.println(s+" \t-> is an operator");
         }
         //operator, identifier, constants
         //variables
         else if(s.matches("[a-z]")){
            System.out.println(s+" \t-> is an identifier");
         }
         else if(s.matches("0")|| s.matches("1")|| s.matches("2") || s.matches("3") || s.matches("4") || s.matches("5") || s.matches("6") || s.matches("7") || s.matches("8") || s.matches("9") || s.matches("10") || s.matches("[A-Z]")){
            System.out.println(s+" \t-> is a constant");
```

```
        }
   switch(s){              case "float":
              System.out.println(s+" \t-> is a keyword");
              break;
           case "char":
              System.out.println(s+" \t-> is a keyword");
              break;
           case "int":
              System.out.println(s+ " \t-> is a keyword");
              break;
           case "const":
              System.out.println(s+ " \t-> is a keyword");
              break;
           case "break":
              System.out.println(s+ " \t-> is a keyword");
              break;
           case "continue":
              System.out.println(s+ " \t---is a keyword");
              break;


           case ";":
              System.out.println(s+" \t-> is a special symbol");
              break;
//    System.out.println("It is a variable")
        }
     }
     }}
```

Output:



Implementation Details

1. Read the high-level language as a source program
2. Convert source programs in to categories of tokens such as Identifiers, Keywords, Constants, Literals and Operators.

Test cases:

1. Input undefined token

Conclusion: Thus we have successfully read a stream of characters as input, and produced the output as a set of tokens. The set of tokens obtained are further useful for syntax analysis phase of the compiler