19CSE401 - Compiler Design

Lab Sheet 5

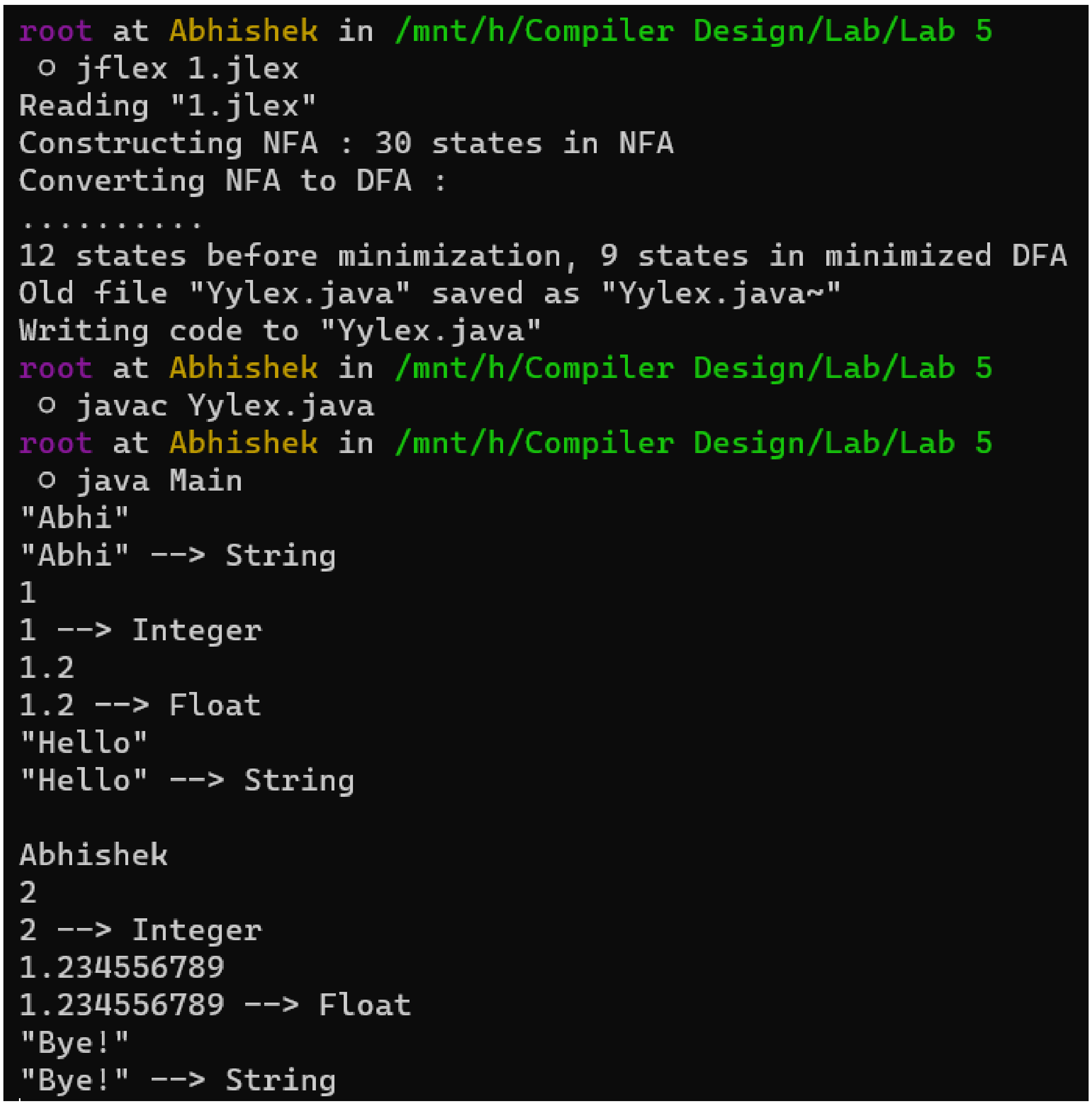
S Abhishek

AM.EN.U4CSE19147

1. Create a lexical generator in JLex to identify the following tokens from the input given in terminal
   a. **String** : set of characters enclosed in "---". Example: "amrita" , "amma123", etc
   b. **Integer** : set of numbers
   c. **Float** : Ex. 0.34, 12.43, 12.0 etc

```
import java.io.*;

class Main
{
    public static void main(String args[]) throws IOException
    {
        Yylex lex=new Yylex(new BufferedReader(new InputStreamReader( System.in )));
        Token token=lex.yylex();

        while(token.text != null )
        {
            token= lex.yylex();
        }
    }
}

class Token
{
    String text;

    Token(String t)
    {
        text = t;
    }
}

%%

Int = [0-9]

%type Token

%eofval{
    return new Token(null);
%eofval}

%%
```

```
\".*\" {System.out.println(yytext() + " --> String");}

{Int}+ {System.out.println(yytext() + " --> Integer");}

{Int}+"."{Int}+ {System.out.println(yytext() + " --> Float");}

[ \t\n]+ {}

. {}
```



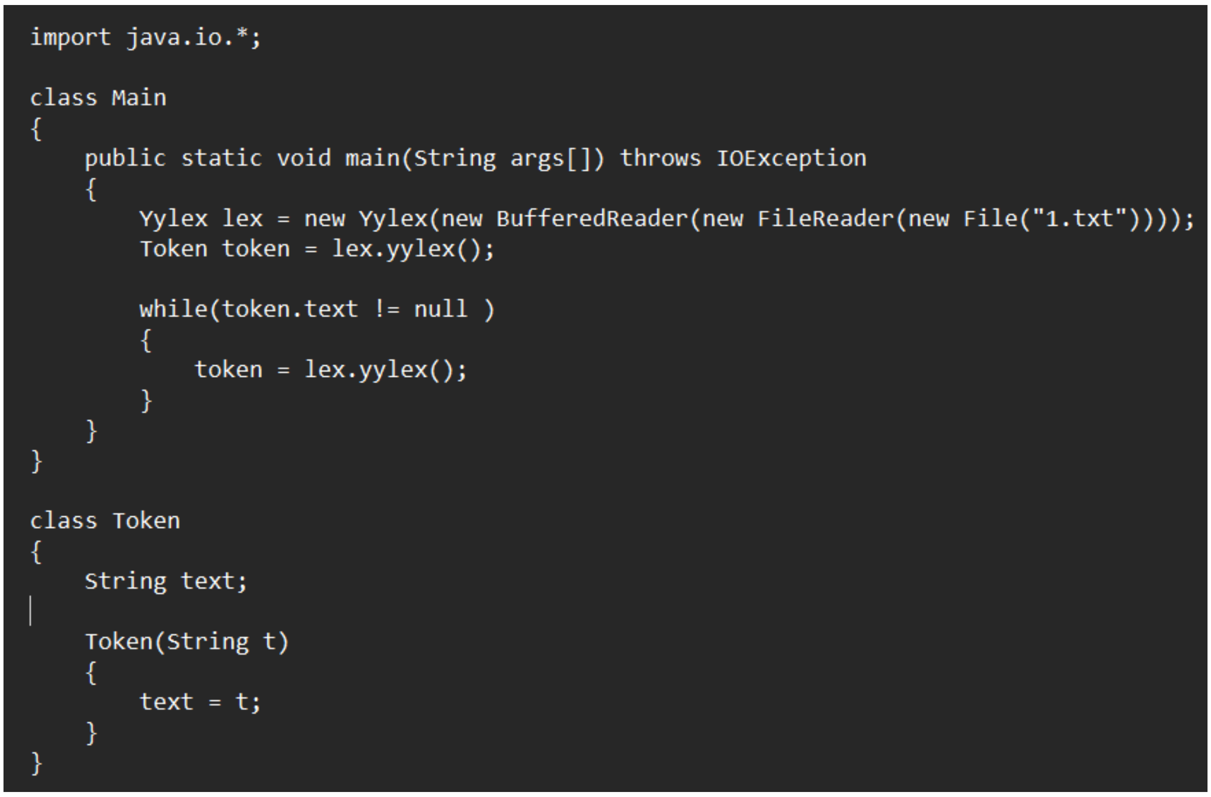
```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 5
 o jflex 1.jlex
Reading "1.jlex"
Constructing NFA : 30 states in NFA
Converting NFA to DFA :
..........
12 states before minimization, 9 states in minimized DFA
Old file "Yylex.java" saved as "Yylex.java~"
Writing code to "Yylex.java"
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 5
 o javac Yylex.java
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 5
 o java Main
"Abhi"
"Abhi" --> String
1
1 --> Integer
1.2
1.2 --> Float
"Hello"
"Hello" --> String

Abhishek
2
2 --> Integer
1.234556789
1.234556789 --> Float
"Bye!"
"Bye!" --> String
```

2. Consider the following token

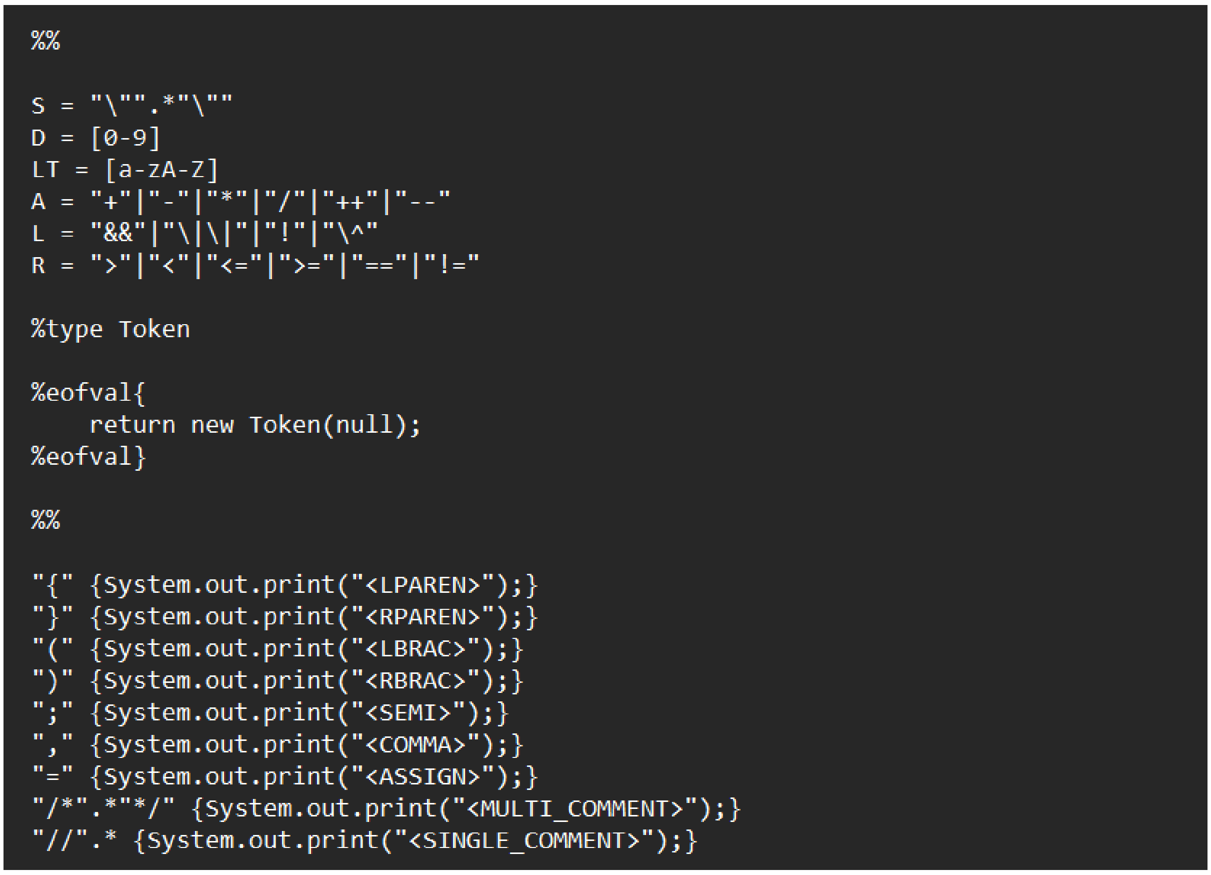
Token	Lexemes		Token	Lexemes
MAIN	main		PRINTF	printf
LPAREN	{		SCANF	scanf
RPAREN	}		RETURN	return
LBRACE	(		INT	int
RBRACE	)		FLOAT	float
ID	For all identifiers		CHAR	char
NUM	Integer constants		/* - -- */	Multiline comment
STR	String Constant		//	Single line comment
REAL	Floating-point constants		SEMI	;
IF	If		COMMA	,
WHILE	while		ARITHMETIC	+, - ,*,/, ++, --
SWITCH	switch		LOGIC	&&, ||, !
CASE	case		RELATIONAL	<, <= , >, >= ==, !=
BREAK	break			

Write a Jlex program that generates the token of the form ***<Token, lexeme>*** except for keywords, for the given program. If none of the patterns matches for a lexeme, given an error statement specifying the line number in the program.

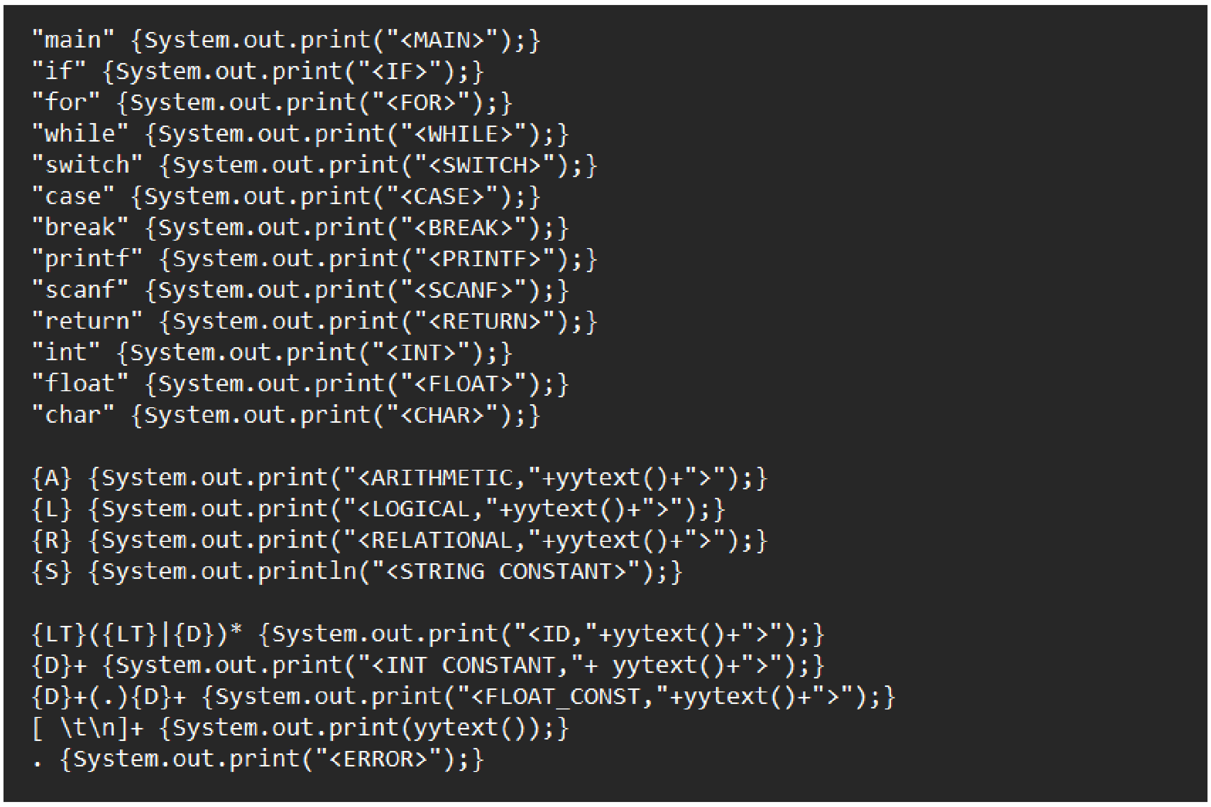
```
int main()
{
 int c, n, f = 1;

 printf("Enter a number to calculate its factorial\n");
 scanf("%d", &n);

 for (c = 1; c <= n; c++)
   f = f * c;

 printf("Factorial of %d = %d\n", n, f);

 return 0;
}
```



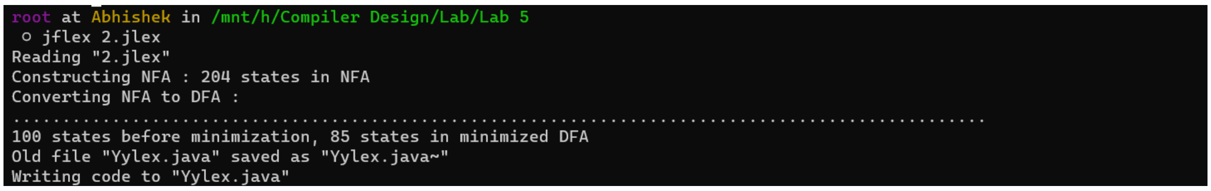
```
import java.io.*;

class Main
{
    public static void main(String args[]) throws IOException
    {
        Yylex lex = new Yylex(new BufferedReader(new FileReader(new File("1.txt"))));
        Token token = lex.yylex();

        while(token.text != null )
        {
            token = lex.yylex();
        }
    }
}

class Token
{
    String text;

    Token(String t)
    {
        text = t;
    }
}
```



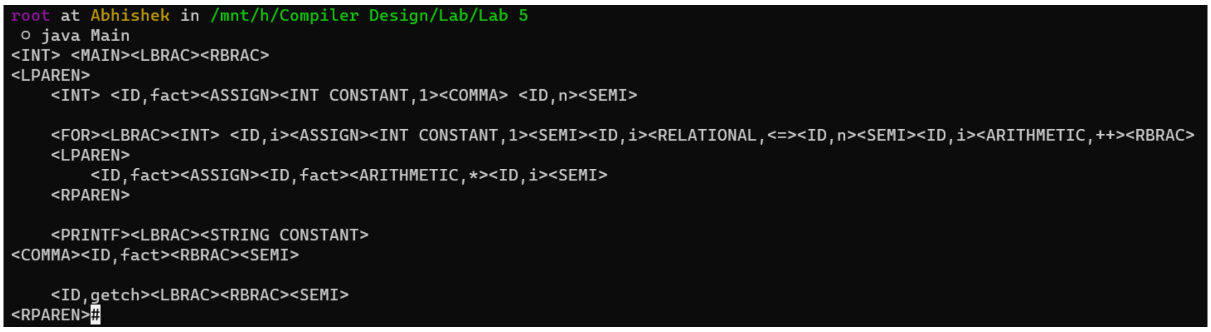
```
%%

S = "\"".*"\""
D = [0-9]
LT = [a-zA-Z]
A = "+"|"-"|"*"|"/"|"++"|"--"
L = "&&"|"\|\|"|"!"|"\^"
R = ">"|"<"|"<="|">="|"=="|"!="

%type Token

%eofval{
    return new Token(null);
%eofval}

%%

"{" {System.out.print("<LPAREN>");}
"}" {System.out.print("<RPAREN>");}
"(" {System.out.print("<LBRAC>");}
")" {System.out.print("<RBRAC>");}
";" {System.out.print("<SEMI>");}
"," {System.out.print("<COMMA>");}
"=" {System.out.print("<ASSIGN>");}
"/*".*"*/" {System.out.print("<MULTI_COMMENT>");}
"//".* {System.out.print("<SINGLE_COMMENT>");}
```



```
"main" {System.out.print("<MAIN>");}
"if" {System.out.print("<IF>");}
"for" {System.out.print("<FOR>");}
"while" {System.out.print("<WHILE>");}
"switch" {System.out.print("<SWITCH>");}
"case" {System.out.print("<CASE>");}
"break" {System.out.print("<BREAK>");}
"printf" {System.out.print("<PRINTF>");}
"scanf" {System.out.print("<SCANF>");}
"return" {System.out.print("<RETURN>");}
"int" {System.out.print("<INT>");}
"float" {System.out.print("<FLOAT>");}
"char" {System.out.print("<CHAR>");}

{A} {System.out.print("<ARITHMETIC,"+yytext()+">");}
{L} {System.out.print("<LOGICAL,"+yytext()+">");}
{R} {System.out.print("<RELATIONAL,"+yytext()+">");}
{S} {System.out.println("<STRING CONSTANT>");}

{LT}({LT}|{D})* {System.out.print("<ID,"+yytext()+">");}
{D}+ {System.out.print("<INT CONSTANT,"+ yytext()+">");}
{D}+(.){D}+ {System.out.print("<FLOAT_CONST,"+yytext()+">");}
[ \t\n]+ {System.out.print(yytext());}
. {System.out.print("<ERROR>");}
```



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 5
 o jflex 2.jlex
Reading "2.jlex"
Constructing NFA : 204 states in NFA
Converting NFA to DFA :
..................................................................................................
100 states before minimization, 85 states in minimized DFA
Old file "Yylex.java" saved as "Yylex.java~"
Writing code to "Yylex.java"
```



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 5
 o java Main
<INT> <MAIN><LBRAC><RBRAC>
<LPAREN>
    <INT> <ID,fact><ASSIGN><INT CONSTANT,1><COMMA> <ID,n><SEMI>

    <FOR><LBRAC><INT> <ID,i><ASSIGN><INT CONSTANT,1><SEMI><ID,i><RELATIONAL,<=><ID,n><SEMI><ID,i><ARITHMETIC,++><RBRAC>
    <LPAREN>
        <ID,fact><ASSIGN><ID,fact><ARITHMETIC,*><ID,i><SEMI>
    <RPAREN>

    <PRINTF><LBRAC><STRING CONSTANT>
<COMMA><ID,fact><RBRAC><SEMI>

    <ID,getch><LBRAC><RBRAC><SEMI>
<RPAREN>#
```

Thankyou!!