19CSE401 - Compiler Design

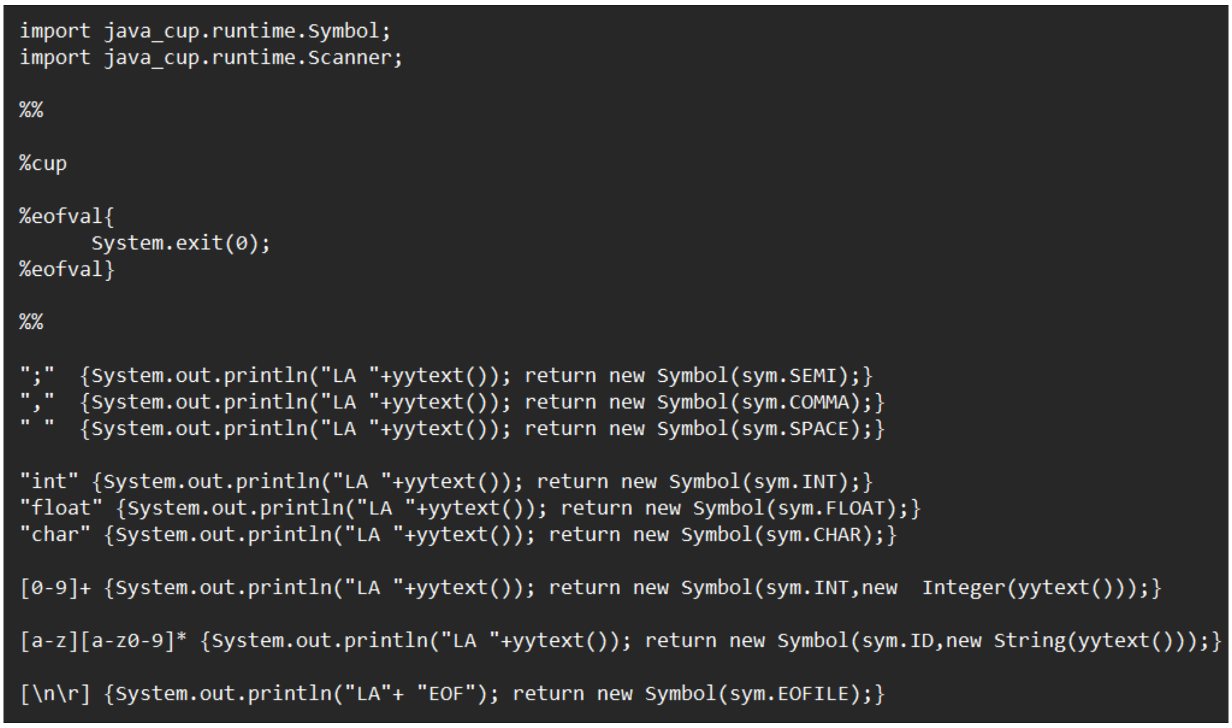
Lab Sheet 6

S Abhishek

AM.EN.U4CSE19147

1. Extend the same grammar to accept the declarations of the form:
    i. int a,b,c;
    ii. float a,b,sum;
    iii. char ch,ch1;

Jlex File



```
import java_cup.runtime.Symbol;
import java_cup.runtime.Scanner;

%%

%cup

%eofval{
      System.exit(0);
%eofval}

%%

";"  {System.out.println("LA "+yytext()); return new Symbol(sym.SEMI);}
","  {System.out.println("LA "+yytext()); return new Symbol(sym.COMMA);}
" "  {System.out.println("LA "+yytext()); return new Symbol(sym.SPACE);}

"int" {System.out.println("LA "+yytext()); return new Symbol(sym.INT);}
"float" {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT);}
"char" {System.out.println("LA "+yytext()); return new Symbol(sym.CHAR);}

[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.INT,new  Integer(yytext()));}

[a-z][a-z0-9]* {System.out.println("LA "+yytext()); return new Symbol(sym.ID,new String(yytext()));}

[\n\r] {System.out.println("LA"+ "EOF"); return new Symbol(sym.EOFILE);}
```

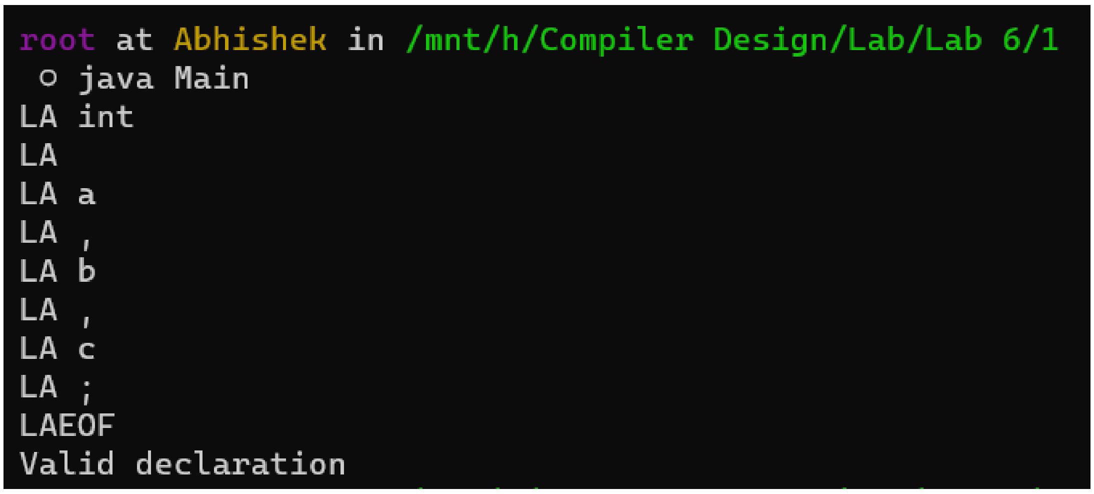
Cup File

```
import java_cup.runtime.*;

scan with {: return getScanner().next_token(); :};


terminal INT, CHAR, FLOAT, SEMI, COMMA, ID, SPACE, EOFILE;

non terminal prog, decln, varlist, type, s;


s ::= prog {: System.out.println("Valid declaration"); :}

EOFILE{:System.exit(0);:};

prog ::=prog decln | decln;

decln ::= type SPACE varlist SEMI;

type ::= INT | CHAR | FLOAT;

varlist ::= ID COMMA varlist | ID ;
```

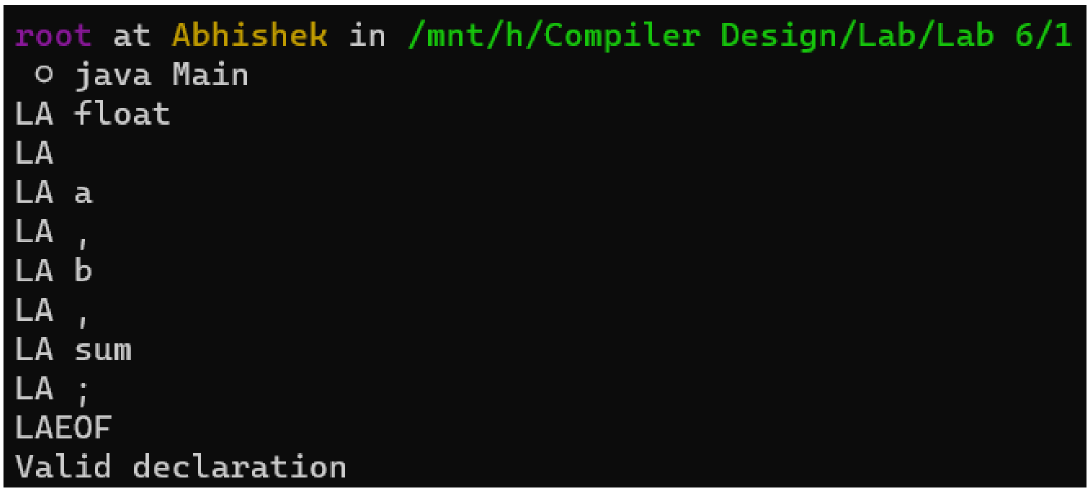
Java File

```
import java.io.*;

public class Main
{
      public static void main(String[] args)throws Exception
      {
            parser p = new parser(new Yylex(new FileReader("Input.txt")));

            p.parse();
      }
}
```

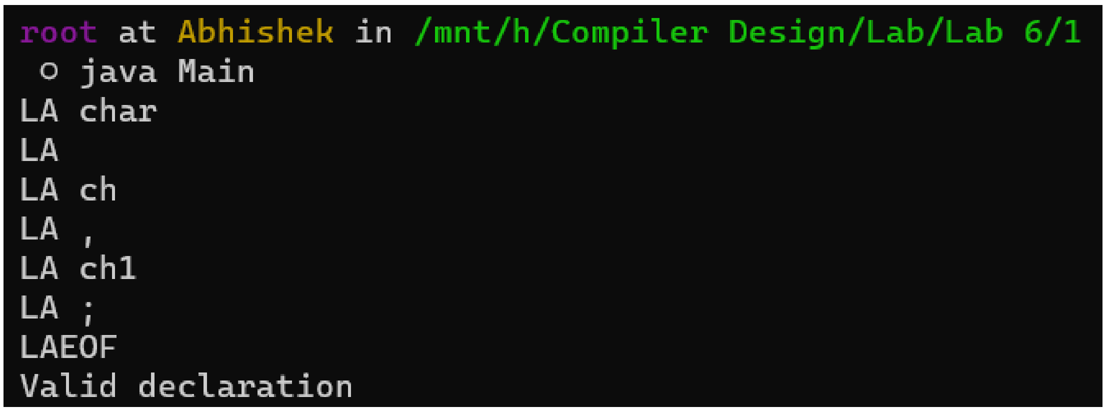
Output



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/1
 ○ java Main
LA int
LA
LA a
LA ,
LA b
LA ,
LA c
LA ;
LAEOF
Valid declaration
```



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/1
 ○ java Main
LA float
LA
LA a
LA ,
LA b
LA ,
LA sum
LA ;
LAEOF
Valid declaration
```

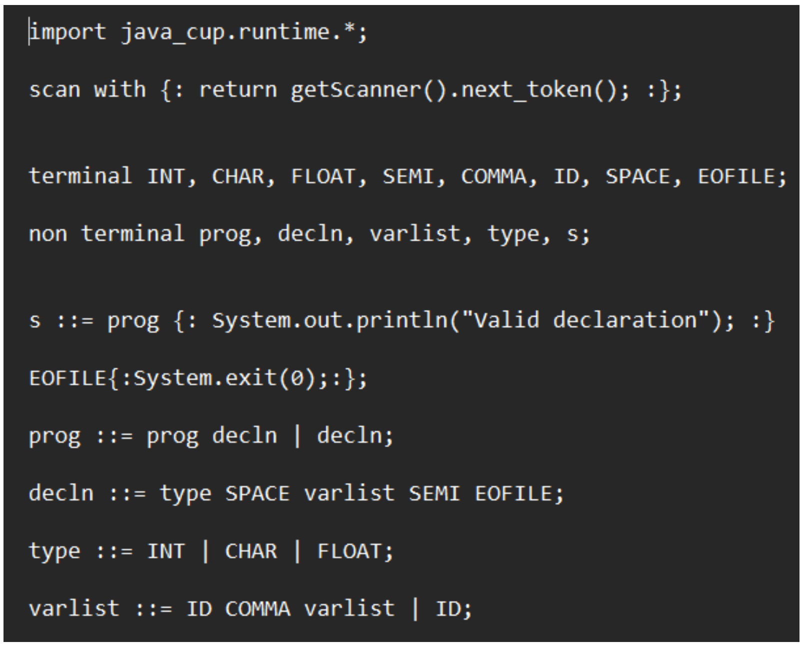


```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/1
 ○ java Main
LA char
LA
LA ch
LA ,
LA ch1
LA ;
LAEOF
Valid declaration
```

2. Given below are some exercises that enhances the simple declaration statement to recognize a wide variety of other declarations.

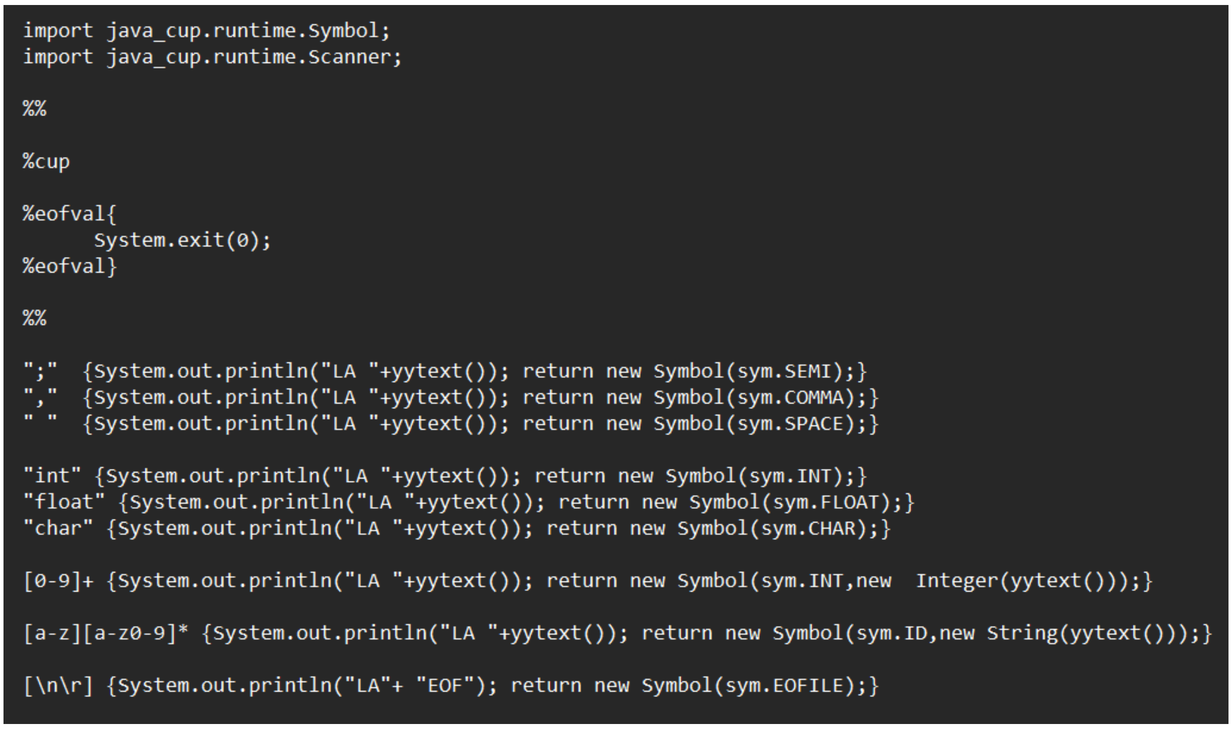
   a. Multiline declaration: Extend the syntax rules to accept declaration statements with newline in between.
      Example: int a,b,c;
                    int c,d;

Cup File



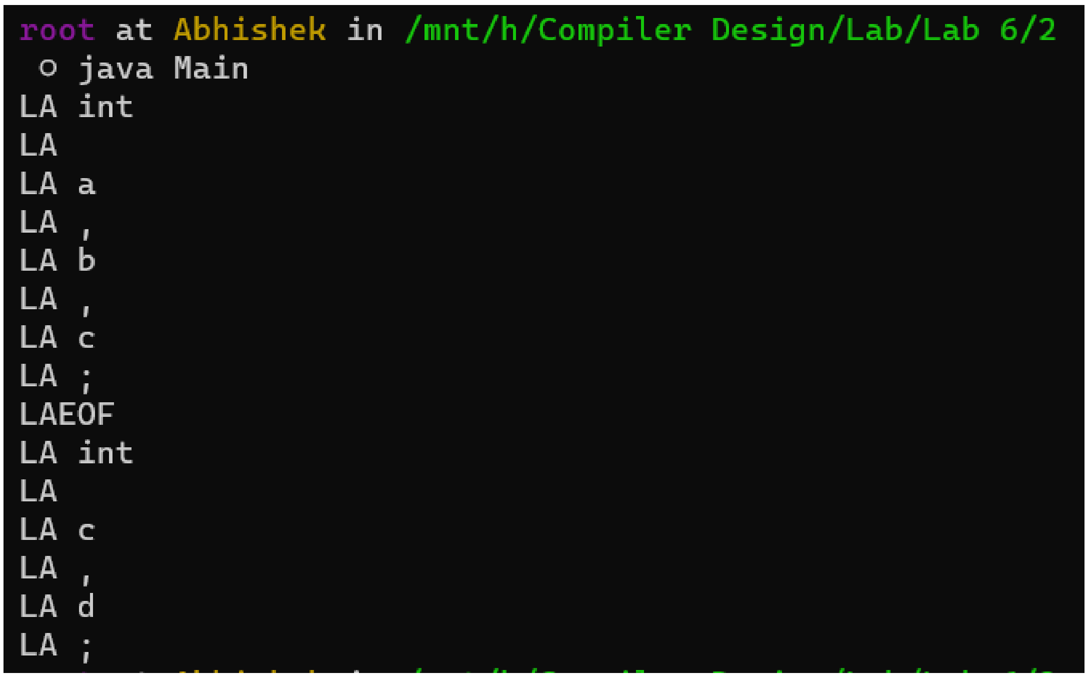
```
import java_cup.runtime.*;

scan with {: return getScanner().next_token(); :};


terminal INT, CHAR, FLOAT, SEMI, COMMA, ID, SPACE, EOFILE;

non terminal prog, decln, varlist, type, s;


s ::= prog {: System.out.println("Valid declaration"); :}

EOFILE{:System.exit(0);:};

prog ::= prog decln | decln;

decln ::= type SPACE varlist SEMI EOFILE;

type ::= INT | CHAR | FLOAT;

varlist ::= ID COMMA varlist | ID;
```

Jlex File



```
import java_cup.runtime.Symbol;
import java_cup.runtime.Scanner;

%%

%cup

%eofval{
      System.exit(0);
%eofval}

%%

";"  {System.out.println("LA "+yytext()); return new Symbol(sym.SEMI);}
","  {System.out.println("LA "+yytext()); return new Symbol(sym.COMMA);}
" "  {System.out.println("LA "+yytext()); return new Symbol(sym.SPACE);}

"int" {System.out.println("LA "+yytext()); return new Symbol(sym.INT);}
"float" {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT);}
"char" {System.out.println("LA "+yytext()); return new Symbol(sym.CHAR);}

[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.INT,new  Integer(yytext()));}

[a-z][a-z0-9]* {System.out.println("LA "+yytext()); return new Symbol(sym.ID,new String(yytext()));}

[\n\r] {System.out.println("LA"+ "EOF"); return new Symbol(sym.EOFILE);}
```

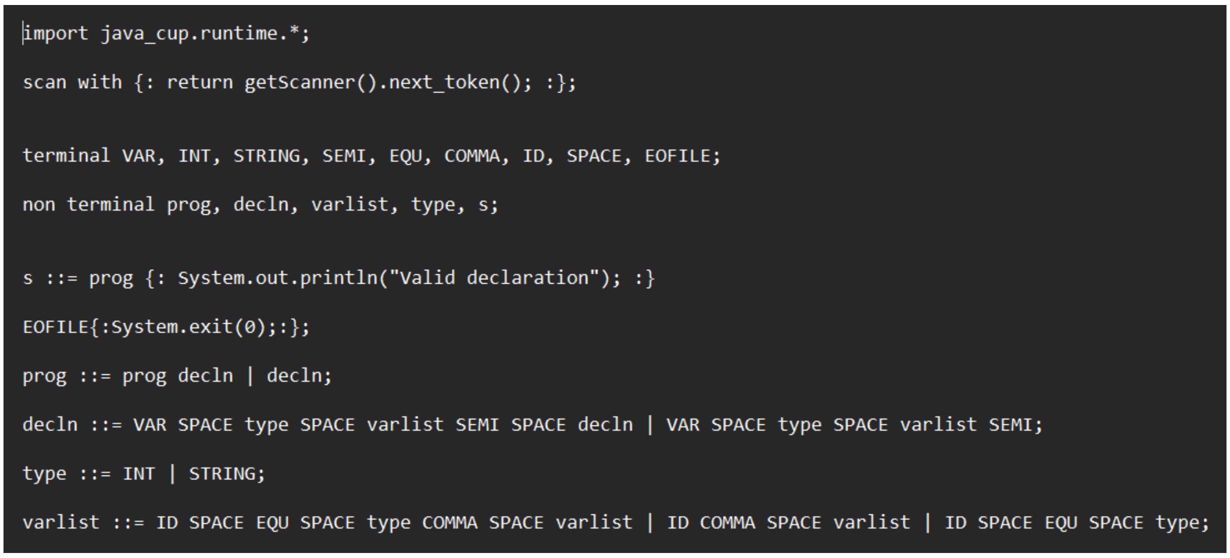
Output



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2
 o java Main
LA int
LA
LA a
LA ,
LA b
LA ,
LA c
LA ;
LAEOF
LA int
LA
LA c
LA ,
LA d
LA ;
```

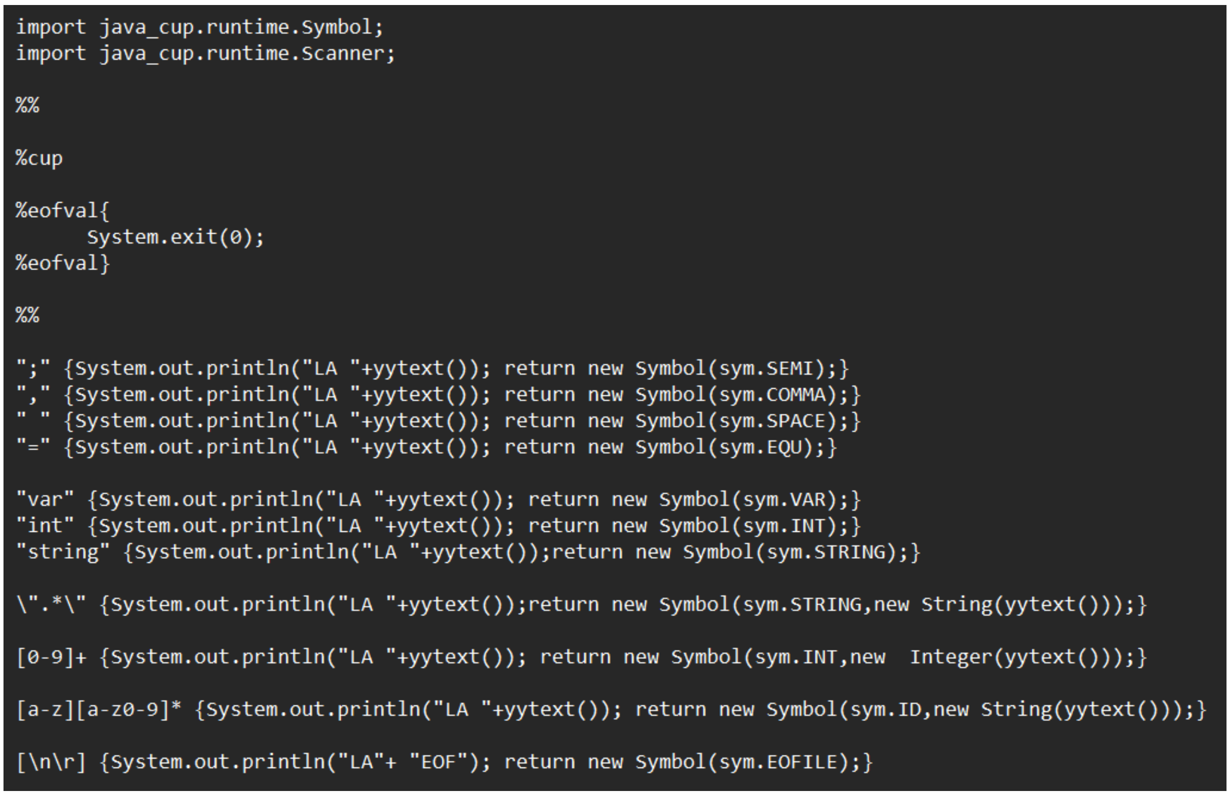
b. Optional Definition: Extend the syntax rules to accept declaration statements of the form: var int tokens = 2, tip, args = 53; var string xyz = “hello”;

Cup File



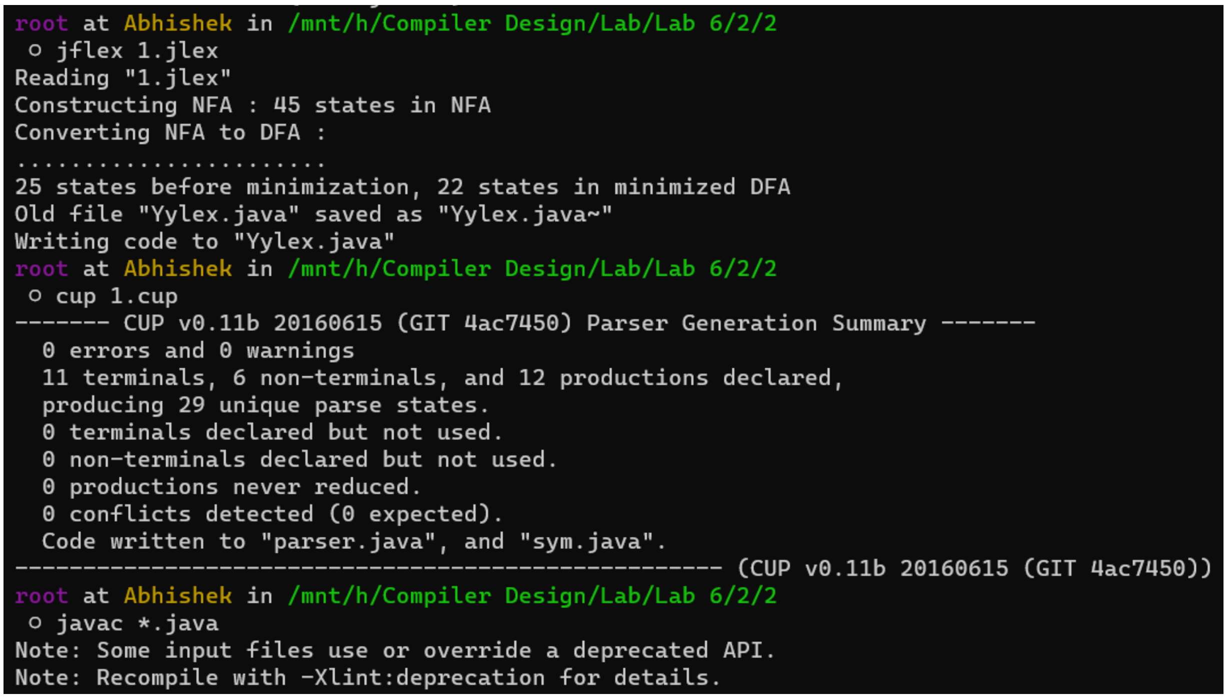
```
import java_cup.runtime.*;

scan with {: return getScanner().next_token(); :};


terminal VAR, INT, STRING, SEMI, EQU, COMMA, ID, SPACE, EOFILE;

non terminal prog, decln, varlist, type, s;


s ::= prog {: System.out.println("Valid declaration"); :}

EOFILE{:System.exit(0);:};

prog ::= prog decln | decln;

decln ::= VAR SPACE type SPACE varlist SEMI SPACE decln | VAR SPACE type SPACE varlist SEMI;

type ::= INT | STRING;

varlist ::= ID SPACE EQU SPACE type COMMA SPACE varlist | ID COMMA SPACE varlist | ID SPACE EQU SPACE type;
```

Jlex File

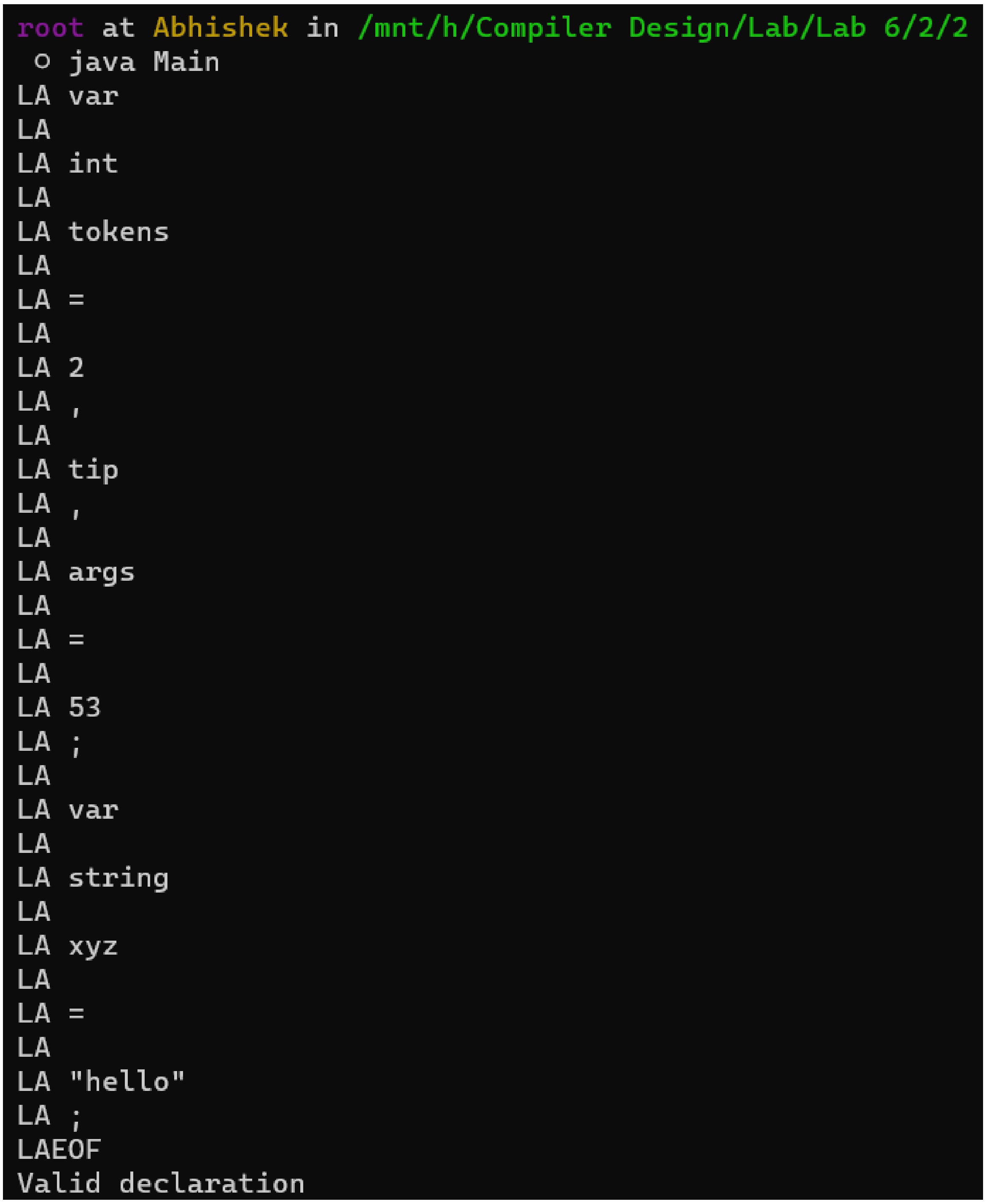


```
import java_cup.runtime.Symbol;
import java_cup.runtime.Scanner;

%%

%cup

%eofval{
      System.exit(0);
%eofval}

%%

";" {System.out.println("LA "+yytext()); return new Symbol(sym.SEMI);}
"," {System.out.println("LA "+yytext()); return new Symbol(sym.COMMA);}
" " {System.out.println("LA "+yytext()); return new Symbol(sym.SPACE);}
"=" {System.out.println("LA "+yytext()); return new Symbol(sym.EQU);}

"var" {System.out.println("LA "+yytext()); return new Symbol(sym.VAR);}
"int" {System.out.println("LA "+yytext()); return new Symbol(sym.INT);}
"string" {System.out.println("LA "+yytext());return new Symbol(sym.STRING);}

\".*\" {System.out.println("LA "+yytext());return new Symbol(sym.STRING,new String(yytext()));}

[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.INT,new  Integer(yytext()));}

[a-z][a-z0-9]* {System.out.println("LA "+yytext()); return new Symbol(sym.ID,new String(yytext()));}

[\n\r] {System.out.println("LA"+ "EOF"); return new Symbol(sym.EOFILE);}
```

Output



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/2
 ○ jflex 1.jlex
Reading "1.jlex"
Constructing NFA : 45 states in NFA
Converting NFA to DFA :
.......................
25 states before minimization, 22 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 6/2/2
 ○ cup 1.cup
------- CUP v0.11b 20160615 (GIT 4ac7450) Parser Generation Summary -------
  0 errors and 0 warnings
  11 terminals, 6 non-terminals, and 12 productions declared,
  producing 29 unique parse states.
  0 terminals declared but not used.
  0 non-terminals declared but not used.
  0 productions never reduced.
  0 conflicts detected (0 expected).
  Code written to "parser.java", and "sym.java".
---------------------------------------------------- (CUP v0.11b 20160615 (GIT 4ac7450))
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/2
 ○ javac *.java
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

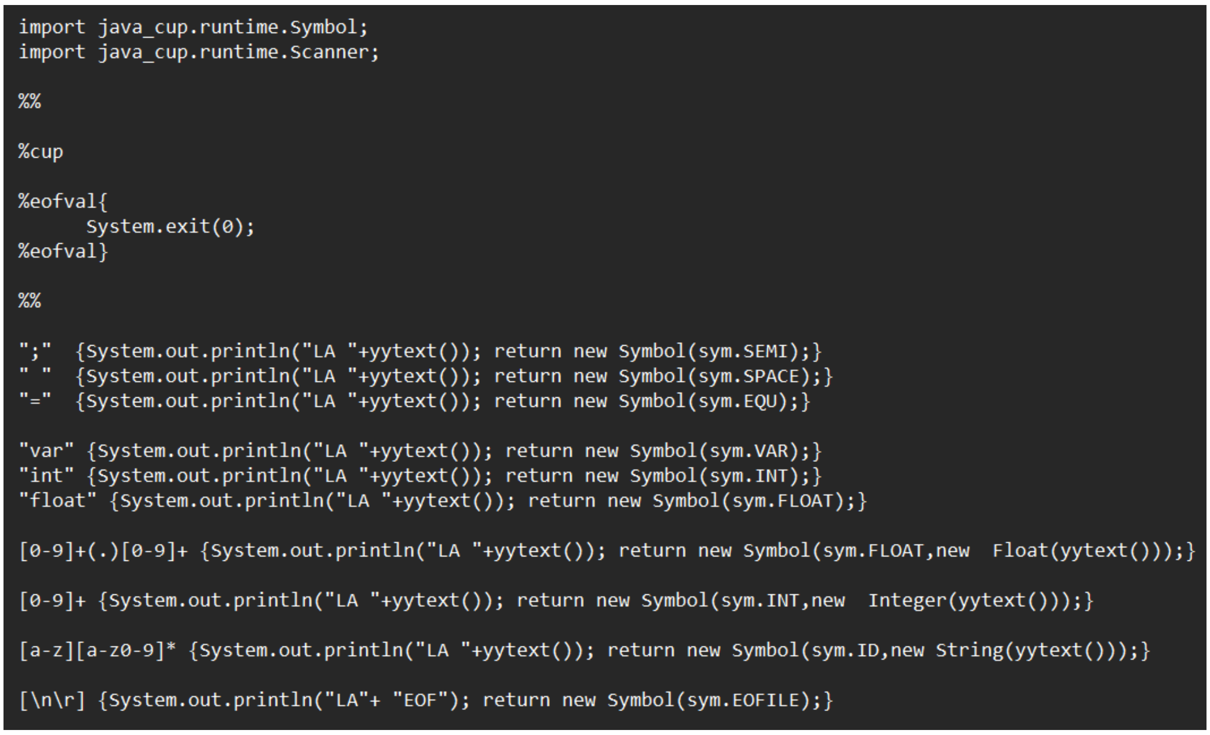


```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/2
 o java Main
LA var
LA
LA int
LA
LA tokens
LA
LA =
LA
LA 2
LA ,
LA
LA tip
LA ,
LA
LA args
LA
LA =
LA
LA 53
LA ;
LA
LA var
LA
LA string
LA
LA xyz
LA
LA =
LA
LA "hello"
LA ;
LAEOF
Valid declaration
```

c. Floating point: Extend the lexical and syntax rules to recognize floating point numbers (2.345) as FLOAT and in declaration statement.

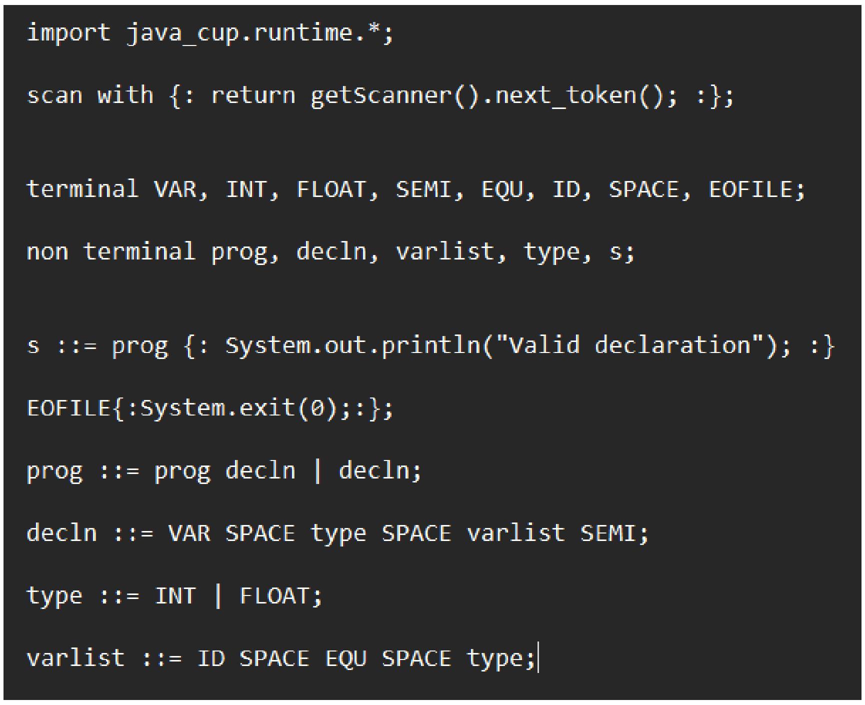
Example: var float time = 12.34;

Jlex File



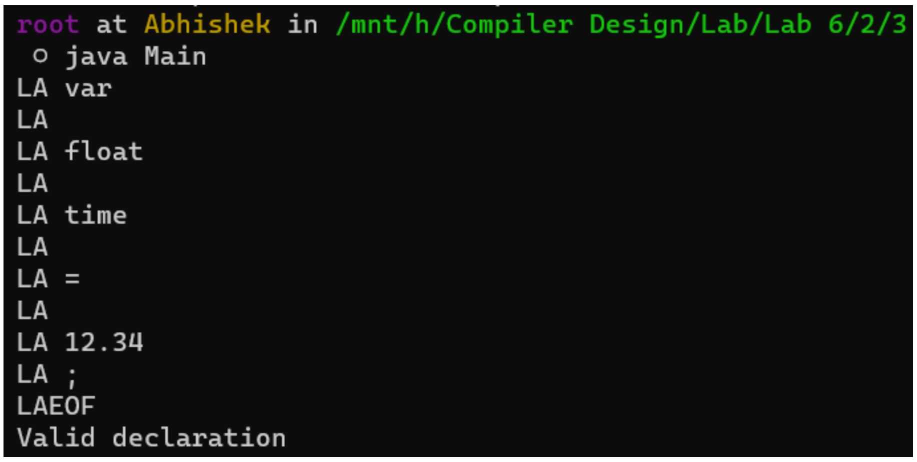
```
import java_cup.runtime.Symbol;
import java_cup.runtime.Scanner;

%%

%cup

%eofval{
      System.exit(0);
%eofval}

%%

";"  {System.out.println("LA "+yytext()); return new Symbol(sym.SEMI);}
" "  {System.out.println("LA "+yytext()); return new Symbol(sym.SPACE);}
"="  {System.out.println("LA "+yytext()); return new Symbol(sym.EQU);}

"var" {System.out.println("LA "+yytext()); return new Symbol(sym.VAR);}
"int" {System.out.println("LA "+yytext()); return new Symbol(sym.INT);}
"float" {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT);}

[0-9]+(.)[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT,new  Float(yytext()));}

[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.INT,new  Integer(yytext()));}

[a-z][a-z0-9]* {System.out.println("LA "+yytext()); return new Symbol(sym.ID,new String(yytext()));}

[\n\r] {System.out.println("LA"+ "EOF"); return new Symbol(sym.EOFILE);}
```

Cup File



```
import java_cup.runtime.*;

scan with {: return getScanner().next_token(); :};


terminal VAR, INT, FLOAT, SEMI, EQU, ID, SPACE, EOFILE;

non terminal prog, decln, varlist, type, s;


s ::= prog {: System.out.println("Valid declaration"); :}

EOFILE{:System.exit(0);:};

prog ::= prog decln | decln;

decln ::= VAR SPACE type SPACE varlist SEMI;

type ::= INT | FLOAT;

varlist ::= ID SPACE EQU SPACE type;
```

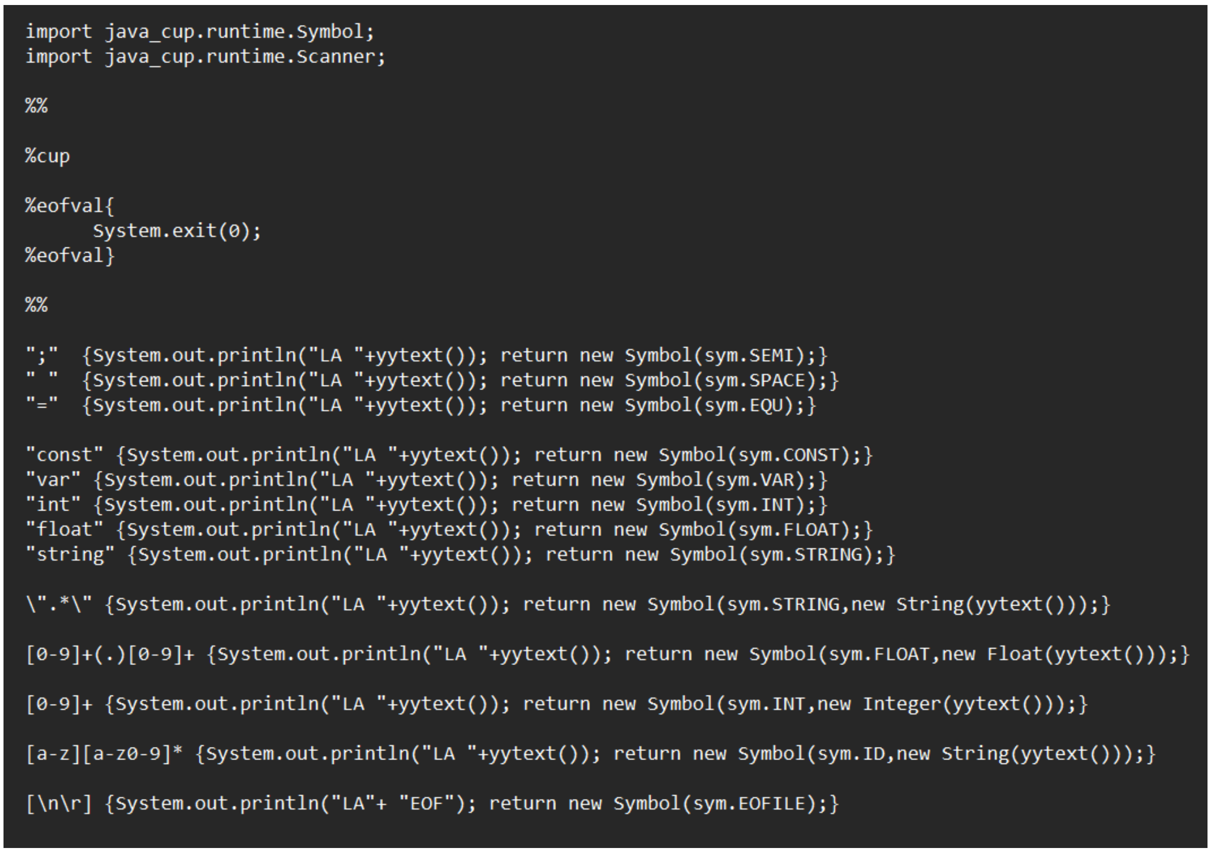
Output



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/3
 o java Main
LA var
LA
LA float
LA
LA time
LA
LA =
LA
LA 12.34
LA ;
LAEOF
Valid declaration
```

d. Constant Declaration: Extend the syntax rules to recognize declaration of constants of the form: const int pi = 3.14;. Note, constant declaration can be done one at a time and definition during declaration is compulsory. i.e. const string xyz; is invalid whereas const string xyz = “hello”; is valid.

Jlex File

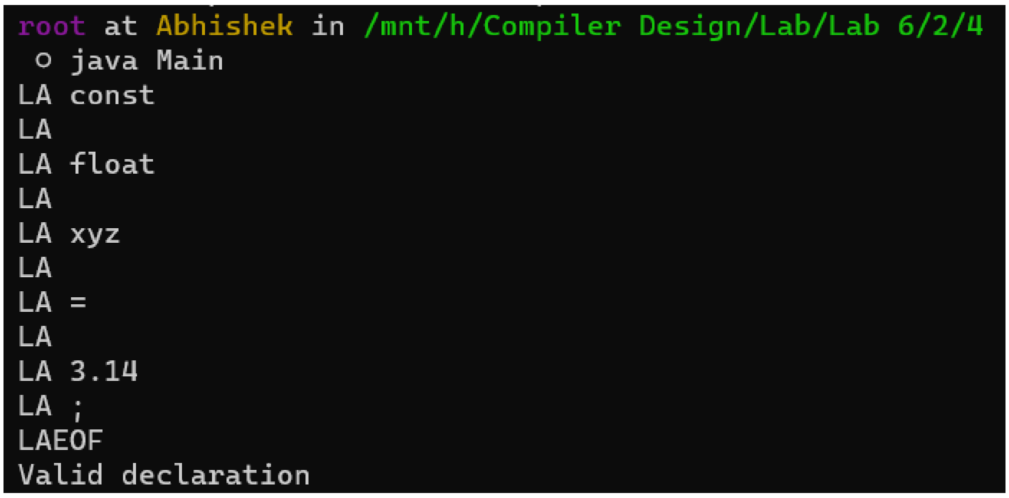


```
import java_cup.runtime.Symbol;
import java_cup.runtime.Scanner;

%%

%cup

%eofval{
      System.exit(0);
%eofval}

%%

";"  {System.out.println("LA "+yytext()); return new Symbol(sym.SEMI);}
" "  {System.out.println("LA "+yytext()); return new Symbol(sym.SPACE);}
"="  {System.out.println("LA "+yytext()); return new Symbol(sym.EQU);}

"const" {System.out.println("LA "+yytext()); return new Symbol(sym.CONST);}
"var" {System.out.println("LA "+yytext()); return new Symbol(sym.VAR);}
"int" {System.out.println("LA "+yytext()); return new Symbol(sym.INT);}
"float" {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT);}
"string" {System.out.println("LA "+yytext()); return new Symbol(sym.STRING);}

\".*\" {System.out.println("LA "+yytext()); return new Symbol(sym.STRING,new String(yytext()));}

[0-9]+(.)[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT,new Float(yytext()));}

[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.INT,new Integer(yytext()));}

[a-z][a-z0-9]* {System.out.println("LA "+yytext()); return new Symbol(sym.ID,new String(yytext()));}

[\n\r] {System.out.println("LA"+ "EOF"); return new Symbol(sym.EOFILE);}
```

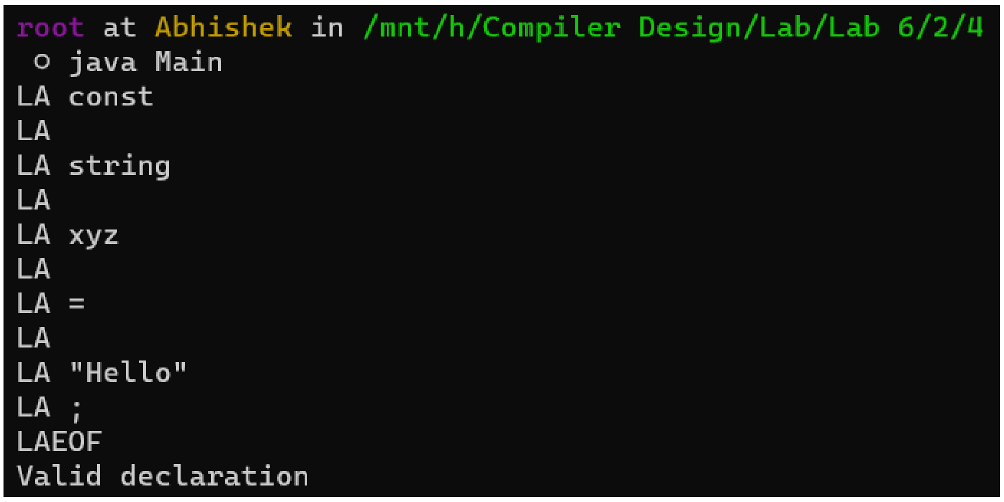
Cup File

```
import java_cup.runtime.*;

scan with {: return getScanner().next_token(); :};


terminal VAR, CONST, INT, FLOAT, STRING, SEMI, EQU, ID, SPACE, EOFILE;

non terminal prog, decln, type, s;


s ::= prog {: System.out.println("Valid declaration"); :}

EOFILE{:System.exit(0);:};

prog ::= prog decln | decln;

decln ::= CONST SPACE type SPACE ID SPACE EQU SPACE type SEMI;

type ::= INT | FLOAT | STRING;
```

Output



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/4
 ○ java Main
LA const
LA
LA float
LA
LA xyz
LA
LA =
LA
LA 3.14
LA ;
LAEOF
Valid declaration
```

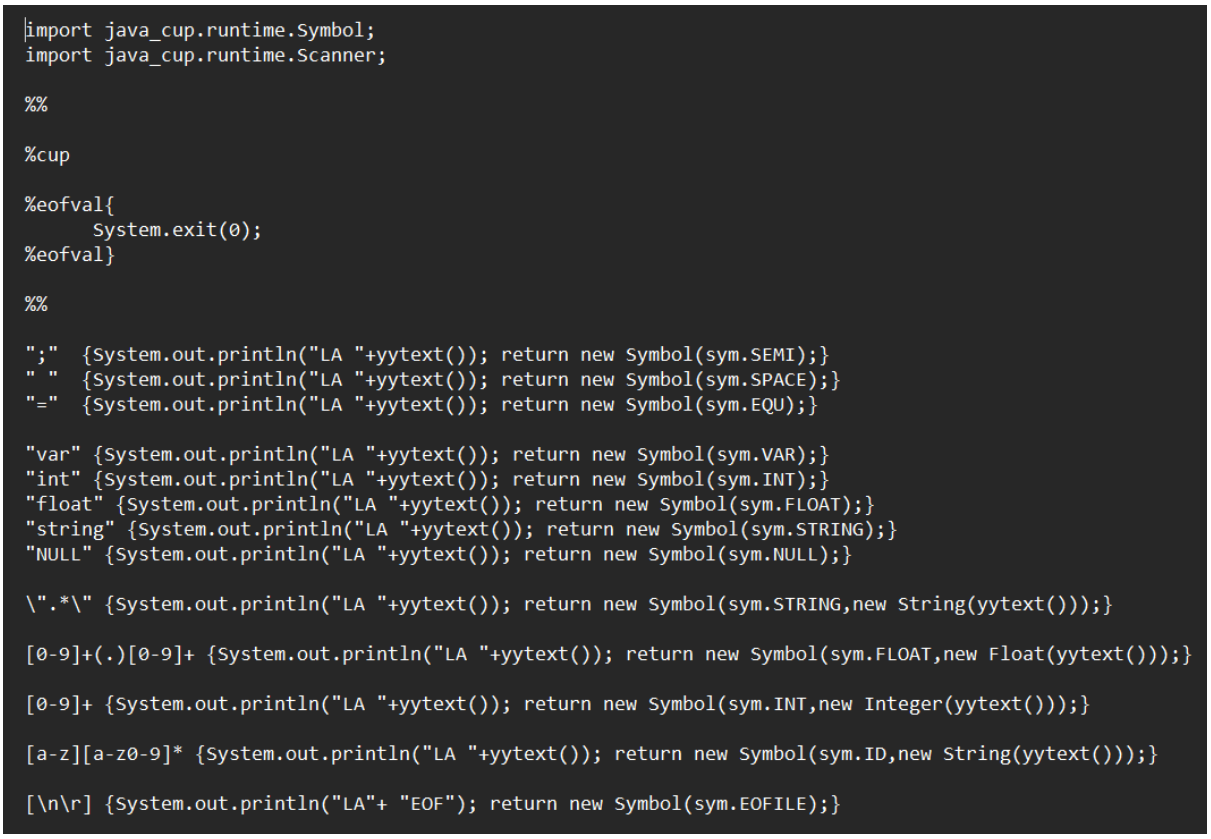


```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/4
 ○ java Main
LA const
LA
LA string
LA
LA xyz
LA
LA =
LA
LA "Hello"
LA ;
LAEOF
Valid declaration
```

e. NULL value: Extend your syntax rules to accept nil as a value during declaration

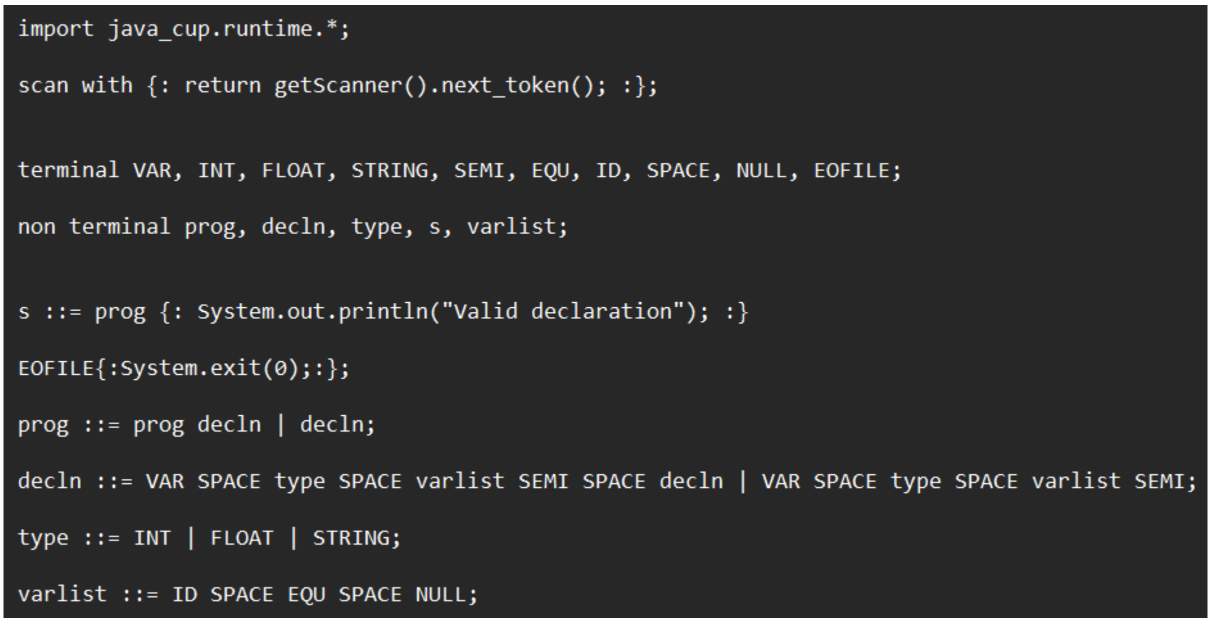
Example: var int i = NULL; var string s = NULL;

Jlex File



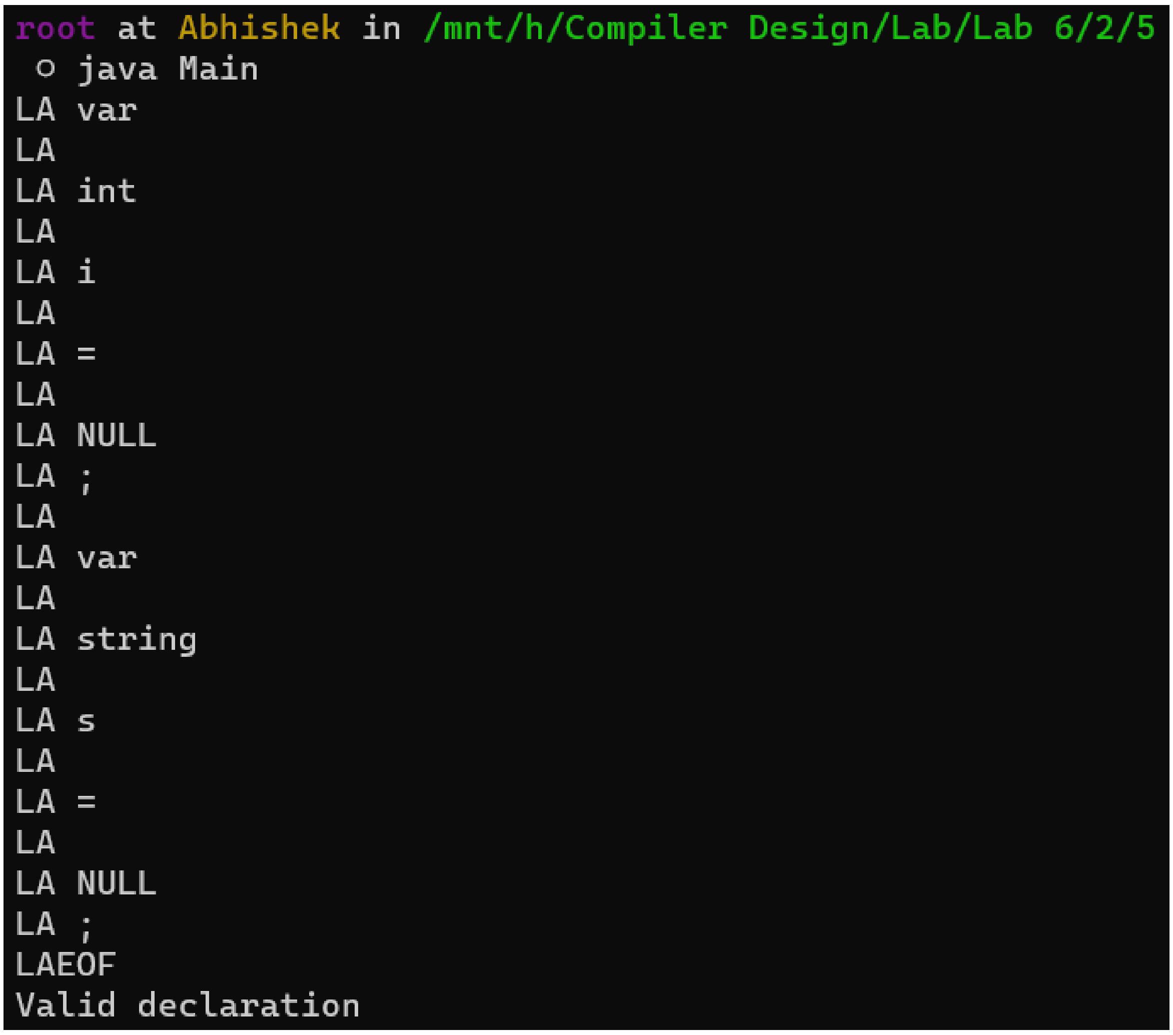
```
import java_cup.runtime.Symbol;
import java_cup.runtime.Scanner;

%%

%cup

%eofval{
      System.exit(0);
%eofval}

%%

";"  {System.out.println("LA "+yytext()); return new Symbol(sym.SEMI);}
" "  {System.out.println("LA "+yytext()); return new Symbol(sym.SPACE);}
"="  {System.out.println("LA "+yytext()); return new Symbol(sym.EQU);}

"var" {System.out.println("LA "+yytext()); return new Symbol(sym.VAR);}
"int" {System.out.println("LA "+yytext()); return new Symbol(sym.INT);}
"float" {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT);}
"string" {System.out.println("LA "+yytext()); return new Symbol(sym.STRING);}
"NULL" {System.out.println("LA "+yytext()); return new Symbol(sym.NULL);}

\".*\" {System.out.println("LA "+yytext()); return new Symbol(sym.STRING,new String(yytext()));}

[0-9]+(.)[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT,new Float(yytext()));}

[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.INT,new Integer(yytext()));}

[a-z][a-z0-9]* {System.out.println("LA "+yytext()); return new Symbol(sym.ID,new String(yytext()));}

[\n\r] {System.out.println("LA"+ "EOF"); return new Symbol(sym.EOFILE);}
```

Cup File



```
import java_cup.runtime.*;

scan with {: return getScanner().next_token(); :};


terminal VAR, INT, FLOAT, STRING, SEMI, EQU, ID, SPACE, NULL, EOFILE;

non terminal prog, decln, type, s, varlist;


s ::= prog {: System.out.println("Valid declaration"); :}

EOFILE{:System.exit(0);:};

prog ::= prog decln | decln;

decln ::= VAR SPACE type SPACE varlist SEMI SPACE decln | VAR SPACE type SPACE varlist SEMI;

type ::= INT | FLOAT | STRING;

varlist ::= ID SPACE EQU SPACE NULL;
```

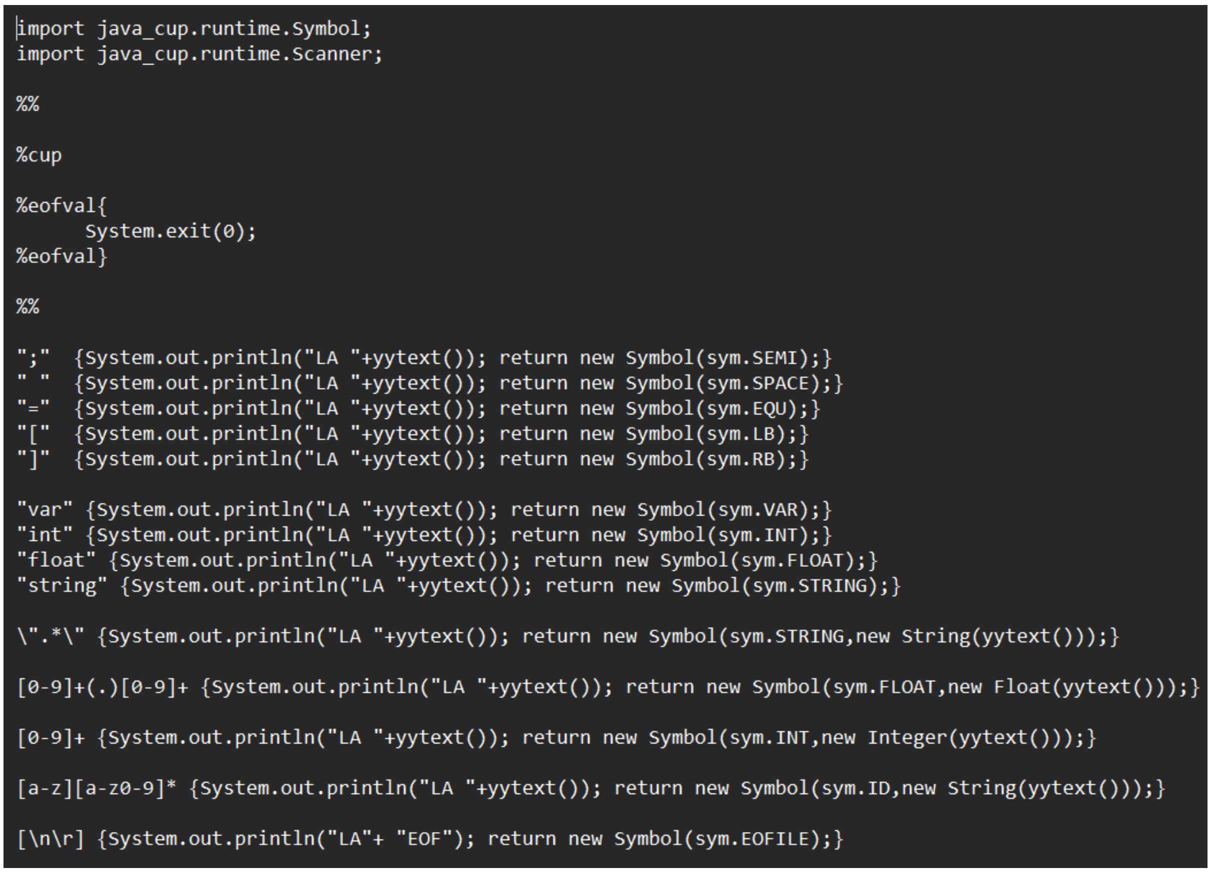
Output



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/5
 ○ java Main
LA var
LA
LA int
LA
LA i
LA
LA =
LA
LA NULL
LA ;
LA
LA var
LA
LA string
LA
LA s
LA
LA =
LA
LA NULL
LA ;
LAEOF
Valid declaration
```

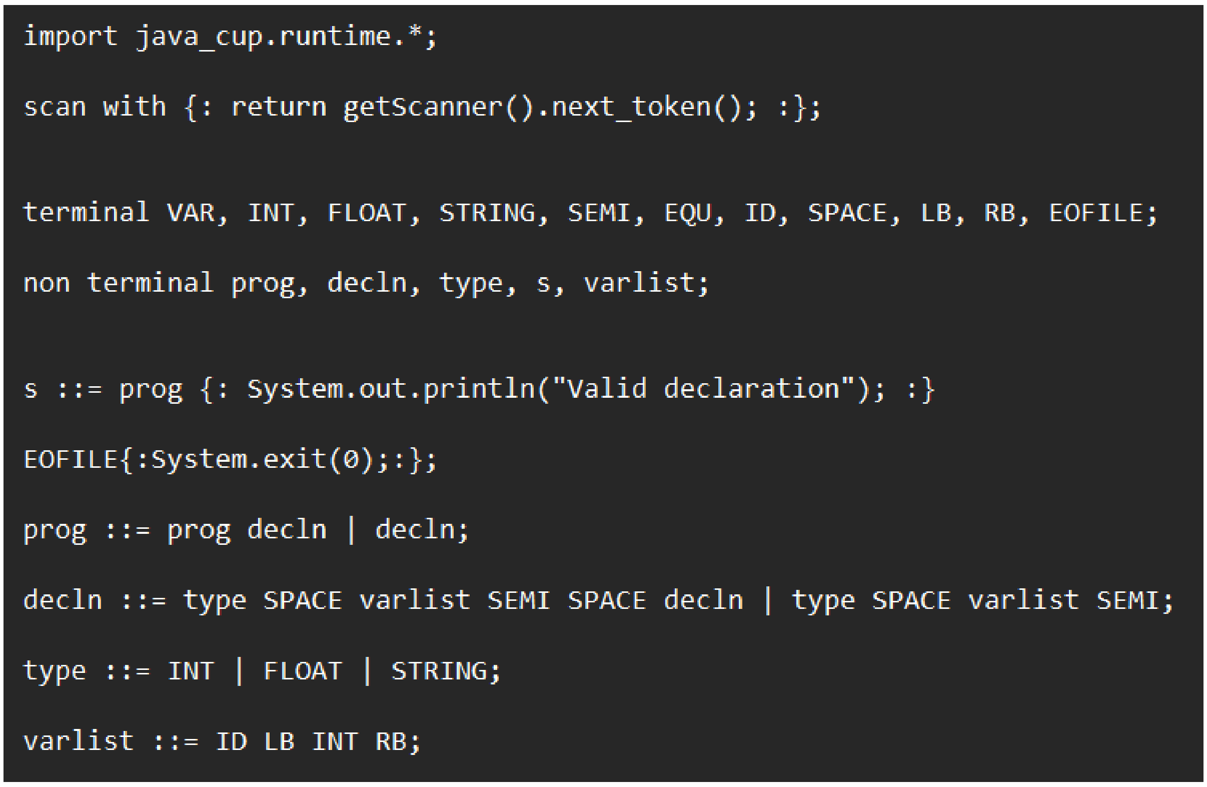
f. Array Declaration: Extend your rules to recognize array declarations of the form: int value[10]; string colours[5];

Jlex File



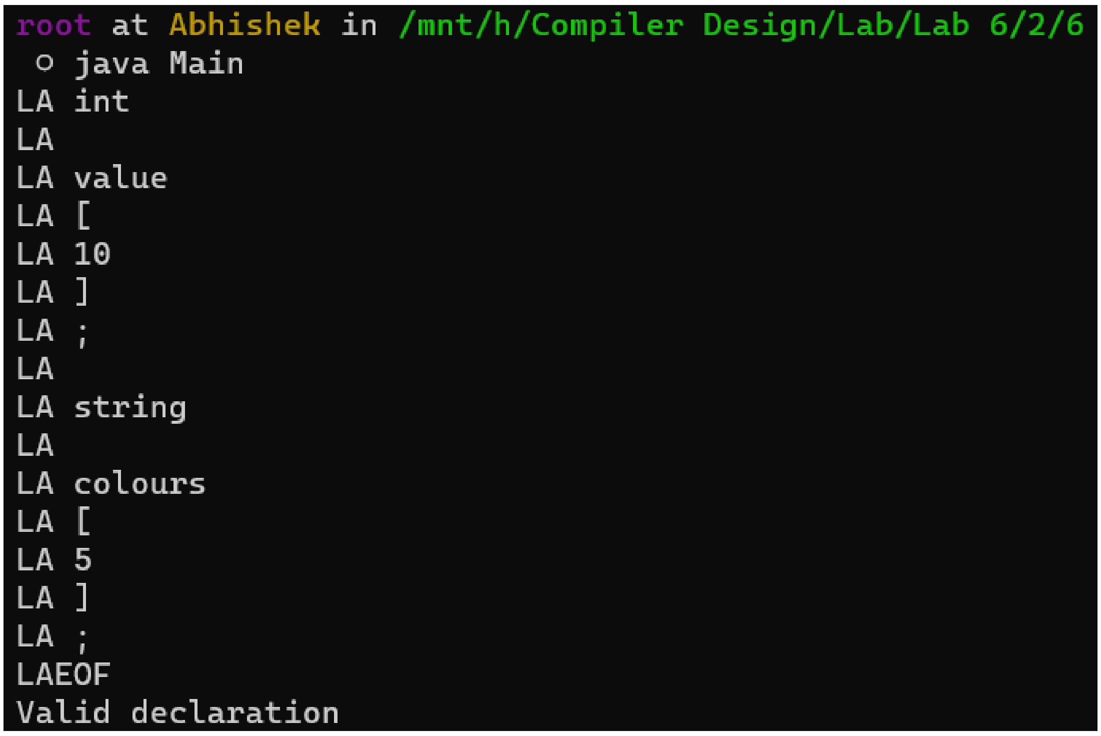
```
import java_cup.runtime.Symbol;
import java_cup.runtime.Scanner;

%%

%cup

%eofval{
      System.exit(0);
%eofval}

%%

";"  {System.out.println("LA "+yytext()); return new Symbol(sym.SEMI);}
" "  {System.out.println("LA "+yytext()); return new Symbol(sym.SPACE);}
"="  {System.out.println("LA "+yytext()); return new Symbol(sym.EQU);}
"["  {System.out.println("LA "+yytext()); return new Symbol(sym.LB);}
"]"  {System.out.println("LA "+yytext()); return new Symbol(sym.RB);}

"var" {System.out.println("LA "+yytext()); return new Symbol(sym.VAR);}
"int" {System.out.println("LA "+yytext()); return new Symbol(sym.INT);}
"float" {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT);}
"string" {System.out.println("LA "+yytext()); return new Symbol(sym.STRING);}

\".*\" {System.out.println("LA "+yytext()); return new Symbol(sym.STRING,new String(yytext()));}

[0-9]+(.)[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT,new Float(yytext()));}

[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.INT,new Integer(yytext()));}

[a-z][a-z0-9]* {System.out.println("LA "+yytext()); return new Symbol(sym.ID,new String(yytext()));}

[\n\r] {System.out.println("LA"+ "EOF"); return new Symbol(sym.EOFILE);}
```

Cup File



```
import java_cup.runtime.*;

scan with {: return getScanner().next_token(); :};


terminal VAR, INT, FLOAT, STRING, SEMI, EQU, ID, SPACE, LB, RB, EOFILE;

non terminal prog, decln, type, s, varlist;


s ::= prog {: System.out.println("Valid declaration"); :}

EOFILE{:System.exit(0);:};

prog ::= prog decln | decln;

decln ::= type SPACE varlist SEMI SPACE decln | type SPACE varlist SEMI;

type ::= INT | FLOAT | STRING;

varlist ::= ID LB INT RB;
```

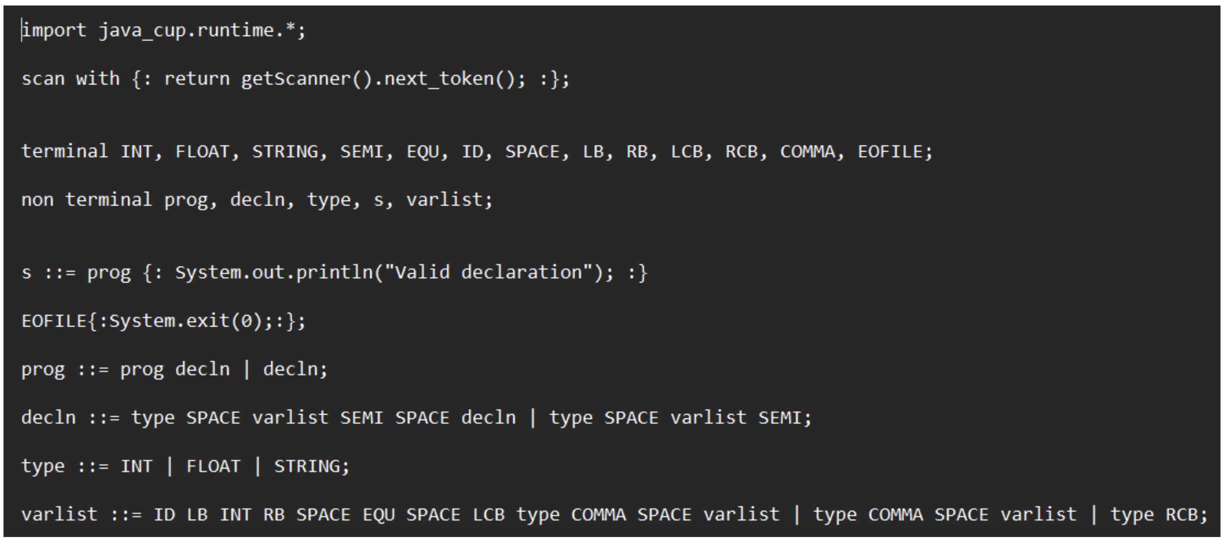
Output



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/6
 o java Main
LA int
LA
LA value
LA [
LA 10
LA ]
LA ;
LA
LA string
LA
LA colours
LA [
LA 5
LA ]
LA ;
LAEOF
Valid declaration
```

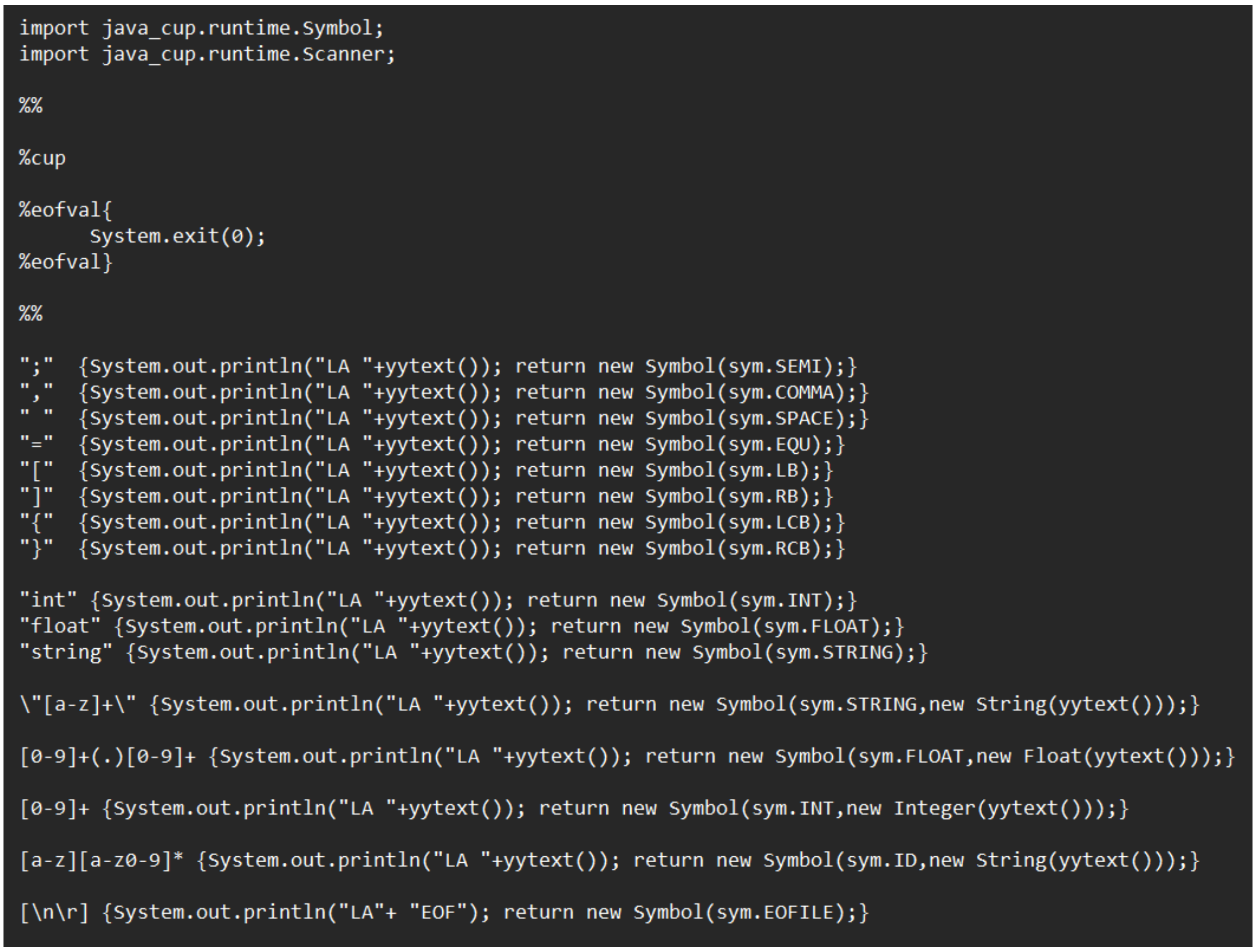
f. Array Initialization: Extend your rules to recognize array initializations of the form: int value[10] = {1, 25, 43, …… }; string colors[5] = {"red", "green", "blue", "orange", "yellow"};

Cup File



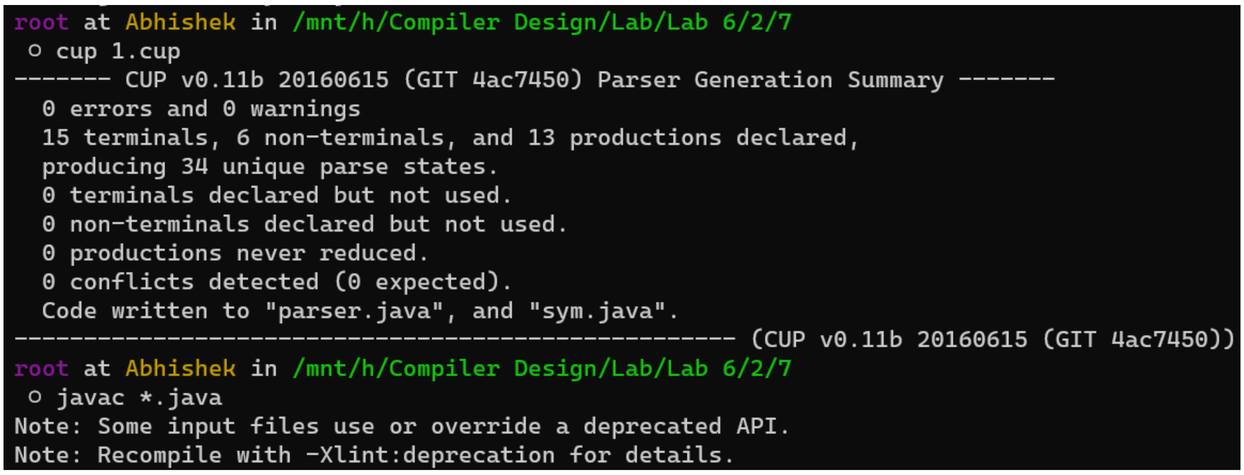
```
import java_cup.runtime.*;

scan with {: return getScanner().next_token(); :};


terminal INT, FLOAT, STRING, SEMI, EQU, ID, SPACE, LB, RB, LCB, RCB, COMMA, EOFILE;

non terminal prog, decln, type, s, varlist;


s ::= prog {: System.out.println("Valid declaration"); :}

EOFILE{:System.exit(0);:};

prog ::= prog decln | decln;

decln ::= type SPACE varlist SEMI SPACE decln | type SPACE varlist SEMI;

type ::= INT | FLOAT | STRING;

varlist ::= ID LB INT RB SPACE EQU SPACE LCB type COMMA SPACE varlist | type COMMA SPACE varlist | type RCB;
```

Jlex File

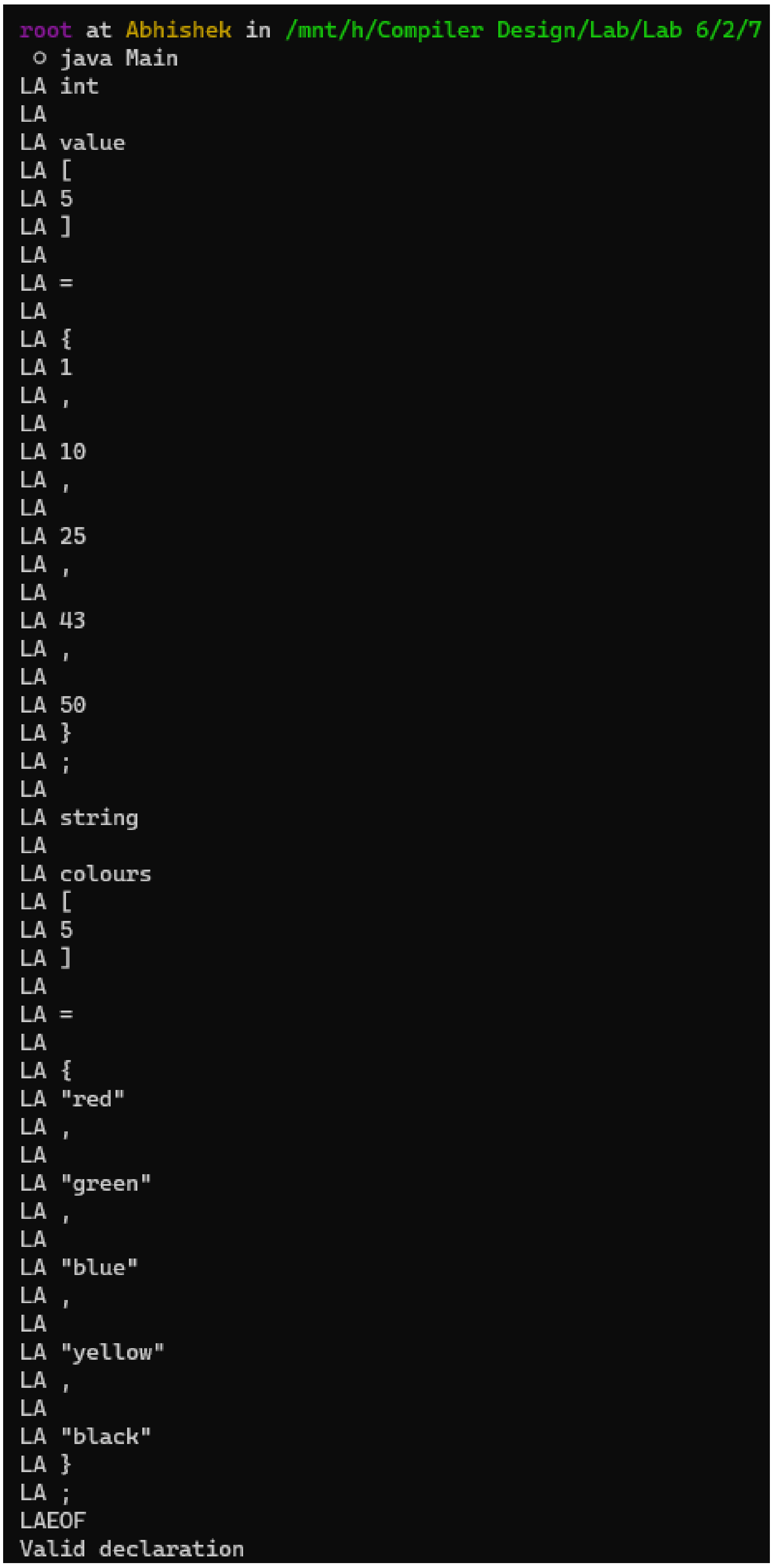


```
import java_cup.runtime.Symbol;
import java_cup.runtime.Scanner;

%%

%cup

%eofval{
      System.exit(0);
%eofval}

%%

";"  {System.out.println("LA "+yytext()); return new Symbol(sym.SEMI);}
","  {System.out.println("LA "+yytext()); return new Symbol(sym.COMMA);}
" "  {System.out.println("LA "+yytext()); return new Symbol(sym.SPACE);}
"="  {System.out.println("LA "+yytext()); return new Symbol(sym.EQU);}
"["  {System.out.println("LA "+yytext()); return new Symbol(sym.LB);}
"]"  {System.out.println("LA "+yytext()); return new Symbol(sym.RB);}
"{"  {System.out.println("LA "+yytext()); return new Symbol(sym.LCB);}
"}"  {System.out.println("LA "+yytext()); return new Symbol(sym.RCB);}

"int" {System.out.println("LA "+yytext()); return new Symbol(sym.INT);}
"float" {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT);}
"string" {System.out.println("LA "+yytext()); return new Symbol(sym.STRING);}

\"[a-z]+\" {System.out.println("LA "+yytext()); return new Symbol(sym.STRING,new String(yytext()));}

[0-9]+(.)[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.FLOAT,new Float(yytext()));}

[0-9]+ {System.out.println("LA "+yytext()); return new Symbol(sym.INT,new Integer(yytext()));}

[a-z][a-z0-9]* {System.out.println("LA "+yytext()); return new Symbol(sym.ID,new String(yytext()));}

[\n\r] {System.out.println("LA"+ "EOF"); return new Symbol(sym.EOFILE);}
```

Output



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/7
 ○ cup 1.cup
------- CUP v0.11b 20160615 (GIT 4ac7450) Parser Generation Summary -------
  0 errors and 0 warnings
  15 terminals, 6 non-terminals, and 13 productions declared,
  producing 34 unique parse states.
  0 terminals declared but not used.
  0 non-terminals declared but not used.
  0 productions never reduced.
  0 conflicts detected (0 expected).
  Code written to "parser.java", and "sym.java".
---------------------------------------------------- (CUP v0.11b 20160615 (GIT 4ac7450))
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/7
 ○ javac *.java
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```



```
root at Abhishek in /mnt/h/Compiler Design/Lab/Lab 6/2/7
 ○ java Main
LA int
LA
LA value
LA [
LA 5
LA ]
LA
LA =
LA
LA {
LA 1
LA ,
LA
LA 10
LA ,
LA
LA 25
LA ,
LA
LA 43
LA ,
LA
LA 50
LA }
LA ;
LA
LA string
LA
LA colours
LA [
LA 5
LA ]
LA
LA =
LA
LA {
LA "red"
LA ,
LA
LA "green"
LA ,
LA
LA "blue"
LA ,
LA
LA "yellow"
LA ,
LA
LA "black"
LA }
LA ;
LAEOF
Valid declaration
```

Thankyou!!