CD LAB 3 Implementation of Syntax Checker for subset of C language using LEX and YACC.

Bhushan Sonawane
BE E 66 (BATCH 3)

```
%{
#include <stdio.h>
extern lineno;
void yyerror(char const *s){}
%}

%token ID INT FLOAT CHAR BLANK IF ELSE WHILE FOR WHILE MAIN CONST

%%
Main:Type MAIN '(' ')' '{' Program '}' {printf("MAIN DETECTED\n");}
 ;

Program:Program SList
     |
;

SList:  SList Declare ';'
   |SList AssignMent ';'
         |SList If
         |SList IfElse
         |SList While
         |SList For '{' SList  {printf("Error at Line:%d - Missing } For Loop\n",lineno);}
         |SList For
         |SList Declare  {printf("Error at Line:%d - Declaration Missing Semicolon \n",lineno);}
   |SList AssignMent  {printf("Error at Line:%d -Assingment Missing Semicolon \n",lineno);}
         |
;
Declare:Type Idlist   {printf("Declare detected \n");}
       |Type AssignMent  {printf("Declarative assignment detected \n");}
;
Type:INT {printf("int detected\n");}
    |FLOAT {printf("float detected\n");}
    |CHAR {printf("char detected\n");}
;

Idlist:Idlist ',' ID {printf("idlist detected\n");}
    |Idlist ',' AssignMent {printf("declarative idlist  assignment detected\n");}
    |ID {printf("id detected\n");}
    |AssignMent {printf("declarative id assignment detected\n");}
;

AssignMent:ID '=' Expression  {printf("idlist with = assignment detected\n");}

;
NumOperator:'+'
   |'-'
   |'*'
```

CD LAB 3 Implementation of Syntax Checker for subset of C language using LEX and YACC.

```
   |'/'
;
Expression: ID NumOperator ID {printf("idlist with + assignment detected\n");}
       | ID
       | CONST
;

Condition:Expression RelOperator Expression
   |Expression
   |
;
RelOperator: '>'
   | '<'
;
StatementBlock: '{' SList {printf("Error at Line:%d - Missing } Bracket \n",lineno);}
   | '{' SList '}'
;
If:IF '(' Condition  {printf("Error at line:%d -Missing ) Bracket\n",lineno);}
   |IF '(' Condition ')' StatementBlock {printf("IF Loop detected\n");}
   | error {printf("Error at line:%d \n",lineno); }
;

IfElse:If ELSE StatementBlock {printf("ELSE Loop detected\n");}
   | error {printf("Error at line:%d \n",lineno); }
;

While: WHILE '(' Condition ')' StatementBlock {printf("WHILE Loop detected\n");}
   | error {printf("Error at line:%d \n",lineno); }
;

ForData:Declare
   |AssignMent
   |Expression
   |Condition
   |
;

For: FOR '(' ForData ';' Condition ';' ForData ')' StatementBlock {printf("FOR Loop detected\n");}
   | error {printf("Error at line:%d \n",lineno); }
;

%%

int main(){
        stdin=fopen("in.c","r");
        stdout=fopen("out","w");
        yyparse();
        return 0;
}
```

CD LAB 3 Implementation of Syntax Checker for subset of C language using LEX and YACC.

LEX

```
%{
#include <stdio.h>
#include "y.tab.h"
int lineno = 0;
%}
letter [a-zA-Z]
digit [0-9]
blank [ \b\t]
nline [\n]
%%
"int" {return INT;}
"float" {return FLOAT;}
"char" {return CHAR;}
"if" {return IF;}
"else" {return ELSE;}
"while" {return WHILE;}
"for" {return FOR;}
"main" {return MAIN;}
{letter}* {return ID;}
{digit}+ {return CONST;}
","|";"|"="|"+"|"<"|">"|"("|")"|"{"|"}" { return yytext[0];}

{nline} {++lineno;}
{blank} {}
%%
```

CD LAB 3 Implementation of Syntax Checker for subset of C language using LEX and YACC.

INPUT

```
int main() {
char ab=h,cd;
int a
int hi;
int b=asd;
a=ab + cd;
a = 1
if(a < b){

if(a{
if{

}else{


}

while(a<b){


}
}

}

for(b = a;i < f ;i ){
}
}
```

OUTPUT

```
Error at Line:4 - Declaration Missing Semicolon
Error at Line:8 -Assingment Missing Semicolon
Error at line:10 -Missing ) Bracket
Error at Line:10 - Missing } Bracket
Error at line:10
Error at line:11
```