

SRN:PES2UG22CS275
NAME:Kushagra Agarwal

```
kushagraagarwal@Kushagras-Macbook PES2UG22CS275_CD_Lab %  
kushagraagarwal@Kushagras-Macbook PES2UG22CS275_CD_Lab % ./chance <lab-1_test-1_valid.c  
Valid Syntax  
kushagraagarwal@Kushagras-Macbook PES2UG22CS275_CD_Lab % ./chance <lab-1_test-2_valid.c  
Error:syntax error,line number:33,token:a  
Valid Syntax  
kushagraagarwal@Kushagras-Macbook PES2UG22CS275_CD_Lab % ./chance <lab-1_test-2_invalid.c  
Error:syntax error,line number:6,token:-  
Error:syntax error,line number:11,token::  
Error:syntax error,line number:12,token::  
Error:syntax error,line number:14,token:a  
Error:syntax error,line number:25,token:*  
Error:syntax error,line number:31,token:while  
Error:syntax error,line number:32,token:while  
Valid Syntax  
kushagraagarwal@Kushagras-Macbook PES2UG22CS275_CD_Lab % ./chance <lab-1_test-1_invalid.c  
Error:syntax error,line number:10,token:if  
Error:syntax error,line number:15,token:(  
Error:syntax error,line number:20,token:else  
Error:syntax error,line number:26,token:a  
Error:syntax error,line number:31,token::  
Error:syntax error,line number:40,token:  
kushagraagarwal@Kushagras-Macbook PES2UG22CS275_CD_Lab %
```

Lex file:

```
%{  
#include<stdio.h>  
#include "y.tab.h"  
int yywrap();  
int yylineno;  
%}
```

```
letter [a-zA-Z]  
digit [0-9]  
id (_|{letter})({letter}|{digit})*  
strlit \"\\.\\\"  
opsign [+~]?  
opfrac (\\.{digit})?  
opexponent ([Ee][+~]?{digit}+)?  
number {opsign}{digit}+{opfrac}{opexponent}  
start \\^*  
end \\*V
```

```
%%  
\\V(.*) ;  
\\*(.*\\n)*.*\\*V {};
```

```
int return INT;  
float return FLOAT;  
char return CHAR;  
bool return BOOL;  
double return DOUBLE;  
static return STATIC;  
main return MAIN;  
if return IF;  
else return ELSE;
```

```
for      return FOR;
while    return WHILE;
do       return DO;
break    return BREAK;
#include  return INCLUDE;
"<".(.)".h>" return HEADER;
{id}     return ID;
{number} return VNUM;
{strlit} return STRLIT;
```

```
"<"     return LT;
">"     return GT;
">="    return GTE;
"<="    return LTE;
"=="    return EQ;
"!="    return NE;
"++"    return INC;
"--"    return DEC;
"||"    return OR;
"&&"    return AND;
"!"     return LNOT;
"("     return SCOMB;
")"     return ECOMB;
"["     return SSQB;
"]"     return ESQB;
{"      return SCURB;
}"      return ECURB;
\r      ;
[' '\t'] ;
\n      ++yylineno;
.        return *yytext;
```

```
%%
int yywrap()
{

}
```

Parser.y

```
%{  
  
#include<stdio.h>  
#include<stdlib.h>  
int yylex();  
void yyerror(char* s);  
extern int yylineno;  
extern char *yytext;  
%}  
%token INT FLOAT DOUBLE CHAR STATIC ID INCLUDE HEADER MAIN DO WHILE IF ELSE FOR  
BOOL BREAK INC DEC STRLIT VNUM LT GT GTE LTE EQ NE OR AND LNOT SCOMB ECOMB  
SSQB ESQB SCURB ECURB  
%start P  
%%  
  
P : S {printf("Valid Syntax\n");YYACCEPT;}  
  ;  
S : INCLUDE HEADER S  
  | STATIC S  
  | MAINF S  
  | DECLR ';' S  
  | ASSGN ';' S  
  |  
  ;  
  
TYPE : INT | FLOAT | CHAR | BOOL | DOUBLE  
      ;  
  
DECLR : TYPE List_Var | s  
      ;  
  
List_Var : List_Var ',' ID | ID  
          ;  
  
ASSGN : TYPE ID '=' EXPR | ID '=' EXPR | STRLIT  
       ;  
  
EXPR : EXPR RELOP E | E | ID INC | ID DEC | LNOT ID|S  
      ;  
  
RELOP : GTE|LTE|EQ|NE|OR|AND|LT|GT  
      ;  
  
E : E '+' T | E '-' T | T  
  ;  
  
T : T '*' F | T '/' F | F  
  ;  
  
F : SCOMB EXPR ECOMB | ID | VNUM  
  ;  
  
MAINF : TYPE MAIN SCOMB Empty_ListVar ECOMB SCURB Stmt ECURB
```

```

;

Empty_ListVar : List_Var
    |
;

Stmt : SingleStmt Stmt | Block Stmt | BREAK
    |
;

Ifelstmt : SingleStmt Stmt | Block Stmt
    ;

SingleStmt : DECLR ';' | ASSGN ';' | IF SCOMB COND ECOMB Ifelstmt | IF SCOMB COND
ECOMB Ifelstmt ELSE Ifelstmt | LOOP | DO Block WHILE COND ';'
    ;

Block : SCURB Stmt ECURB
    ;

LOOP : WHILE SCOMB COND ECOMB LOOP2
    | FOR SCOMB COND ECOMB LOOP2
    ;

COND : EXPR | ASSGN
    ;

LOOP2 : SCURB Stmt ECURB
    |
    ;

s : error;
%%
void yyerror(char* s)
{
printf("Error:%s,line number:%d,token:%s\n",s,yylineno,yytext);
}

int main()
{
yyvsparse();
}

```