**Code:-**

**Declaration.l**

%{

#include "y.tab.h";

#include<stdio.h>

%}

DIGIT [0-9]

REAL {DIGIT}+["."]{DIGIT}+

LETTERS [a-zA-Z]

IDENTIFIER (\_({LETTERS}|\_)|{LETTERS})({LETTERS}|{DIGIT}|"\_")\*

BOOLEANVAL true|false

CHARVAL ['][a-zA-Z0-9][']

COMMA ,

EQUAL =

SEMICOLON ;

NEWLINE \n

%%

[ \t] ;

int {return INT;}

char {return CHAR;}

boolean {return BOOLEAN;}

float {return FLOAT;}

{DIGIT} {return INTVAL;}

{CHARVAL} {return CHARVAL;}

{BOOLEANVAL} {return BOOLEANVAL;}

{REAL} {return REALVAL;}

{IDENTIFIER} {return IDENTIFIER;}

{EQUAL} {return EQUAL;}

{COMMA} {return COMMA;}

{SEMICOLON} {return SEMICOLON;}

{NEWLINE} {return 0;}

. ;

%%

int yywrap() {

return 1;

}

**Declaration.y**

%{

#include<stdio.h>

%}

%token INT CHAR BOOLEAN FLOAT EQUAL IDENTIFIER SEMICOLON COMMA INTVAL CHARVAL BOOLEANVAL REALVAL;

%%

Declaration : int\_declaration | char\_declaration | boolean\_declaration | float\_declaration;

int\_declaration : INT int\_declaration\_ext SEMICOLON {printf("Valid integer type variable declaration \n");}

char\_declaration : CHAR char\_declaration\_ext SEMICOLON {printf("Valid char type variable declaration \n");}

boolean\_declaration : BOOLEAN boolean\_declaration\_ext SEMICOLON {printf("Valid boolean type variable declaration \n");}

float\_declaration : FLOAT float\_declaration\_ext SEMICOLON {printf("Valid float type variable declaration \n");}

int\_declaration\_ext : IDENTIFIER | IDENTIFIER COMMA int\_declaration\_ext | IDENTIFIER EQUAL INTVAL |IDENTIFIER EQUAL INTVAL COMMA int\_declaration\_ext;

char\_declaration\_ext : IDENTIFIER | IDENTIFIER COMMA char\_declaration\_ext | IDENTIFIER EQUAL CHARVAL |IDENTIFIER EQUAL CHARVAL COMMA char\_declaration\_ext;

boolean\_declaration\_ext : IDENTIFIER | IDENTIFIER COMMA boolean\_declaration\_ext | IDENTIFIER EQUAL BOOLEANVAL |IDENTIFIER EQUAL BOOLEANVAL COMMA boolean\_declaration\_ext;

float\_declaration\_ext : IDENTIFIER | IDENTIFIER COMMA float\_declaration\_ext | IDENTIFIER EQUAL REALVAL |IDENTIFIER EQUAL REALVAL COMMA float\_declaration\_ext;

%%

void yyerror(char\* s) {

printf("ERROR %s",s);

}

int main(int argc, char\*\* argv) {

yyparse();

return 0;

}

**Output:-**

E:\TE\SPOS\Declaration>flex declaration.l

E:\TE\SPOS\Declaration>bison -dy declaration.y

E:\TE\SPOS\Declaration>gcc lex.yy.c y.tab.c -o declaration.exe

E:\TE\SPOS\Declaration>declaration.exe

int a,b,c=5;

Valid integer type variable declaration

E:\TE\SPOS\Declaration>declaration.exe

int a=0.5;

ERROR syntax error