Practical no: 7

Problem Statement: - Write a program using Yacc specifications to implement lexical analysis Phase of compiler to validate type and syntax of variable declaration in Java.

Name: Aditi Dinesh Mulay

Class: T.E. Computer

Subject: SPOS

Div: A

Roll no: 02

PRN No. 71918146B

SPOS

Assignment B-4

Aditi Dinesh Mulay
T.E. Comp Div: A
Roll no: 02

Aim: Design LEX & Yacc program to validate type & syntax of variable declaration in Java.

Problem Statement:

Write a program using Yacc specifications to implement lexical analysis phase of compiler to validate type and syntax of variable declaration in Java.

Pre-requisites: LEX110, LEX120, LEX130, LEX140, LEX160, 250.

Software Requirement: o/s - Ubuntu, Kylin
Slwname - FLEX, YACC

Theory:

YACC is a computer program for the UNIX O.S. developed by Stephan C. Johnson. It is a Look Ahead Left-to-Right parser generator, generating a parser, the part of a compiler that tries to make syntatic sense of source code, specifically a LALR parser, based on an analytic grammer written in a notation similar to Backus-Naur form (BNF). Yacc is offically known as a 'parser'. Its job is to analyze the structure of i/p stream, & operate of big picture.

Structure of yacc file: A yacc file looks much like a lex file:

```
...definitions...
%%%
...rules...
%%%...code...
```

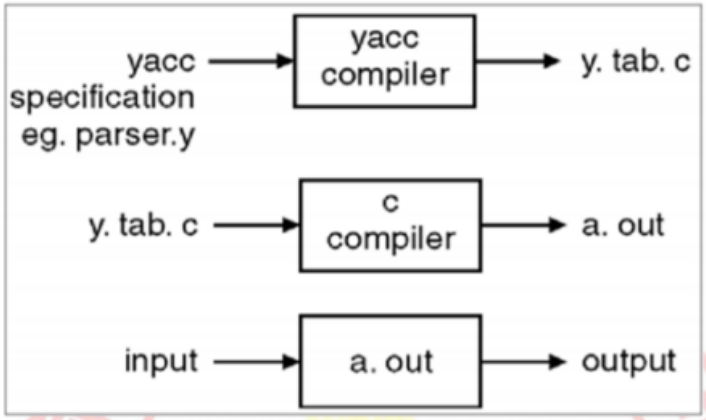


Fig :YACC Parser Generation Model

Most constructs in Modern programming language. can be represeated in BNF. For ex. grammer for an expression that multiplies and adds number is:

1 E → E+E
2 E → E*E
3 E → id.

Translating, Compiling & Executing A Yacc Prg:

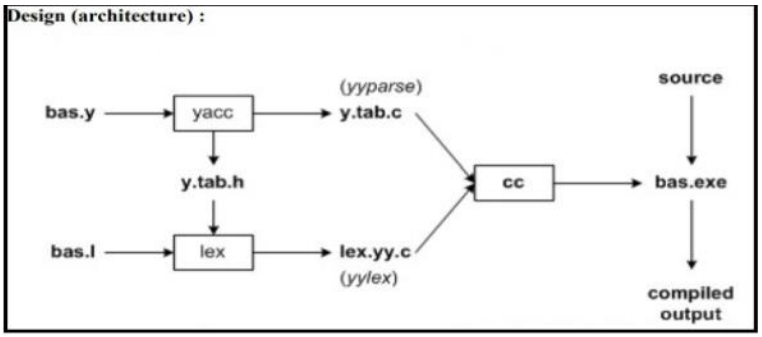
Lex prg. file consist of Lex specifications & should be named .l and the Yacc program consists of Yacc specifications & should be named .y. following command may be issued to generate the parser.

```
lex<filename>.l
Yacc -d<filename>.y
cc lex.yy.c y.tab.c -ll
./a.out
```

Lexical Analyzer for Yacc:

```
yylex(){
extern int yylval;
int c;
...
c = getchar();
...switch(c){
...case '0':
case '1':
...
case '9':
yylval = c - '0';
return (DIGIT);
...}
```

Design (architecture) :



Comparing Sentence Types:

1. The simple sentence is an independent clause with one subject and one verb.
2. The compound sentence is two or more independent clause, joined with comma, semicolon & conjunctions.

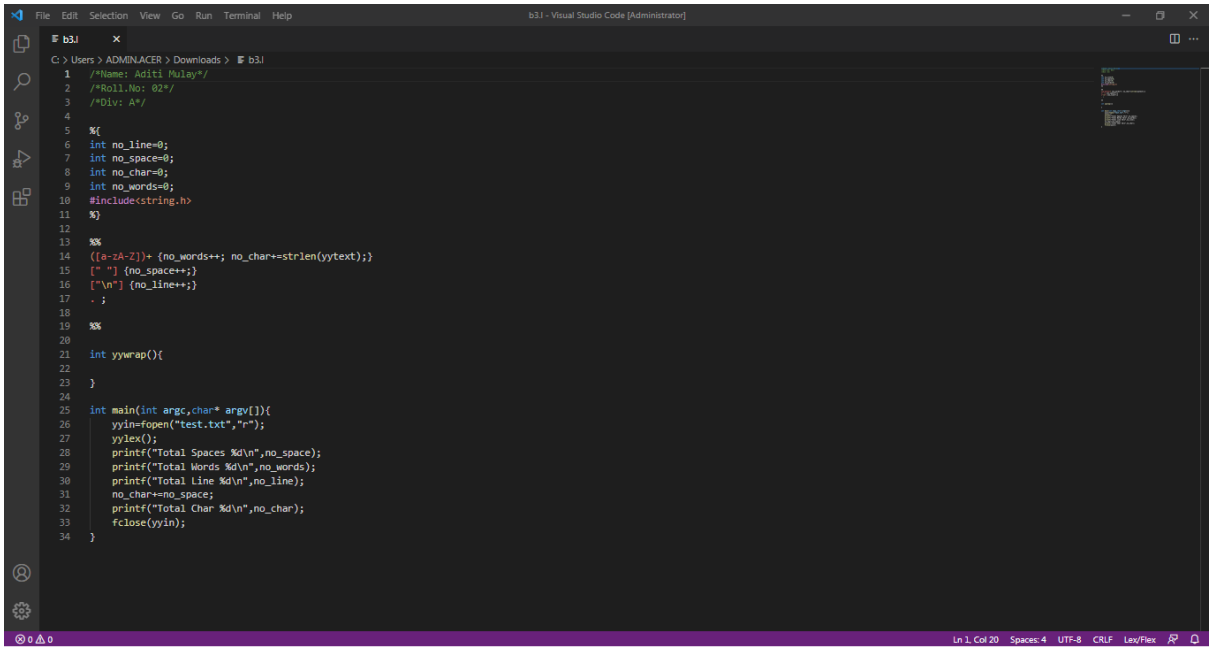
Applications:

YACC is used to generate parsers, which are an integral part of compiler.

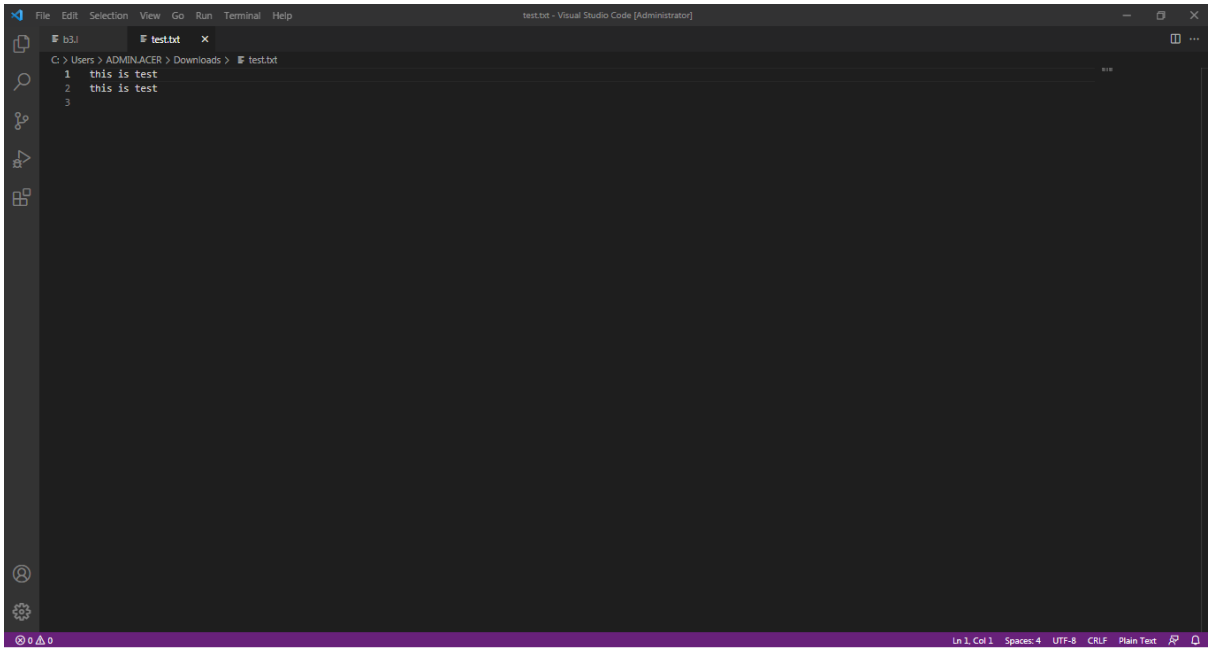
Conclusion:

Thus, we have studied lexical analyzer, syntax analysis and implemented LEX & YACC application of syntax analyzer to validate the given infix expression.

PROGRAM:



```
/*Name: Aditi Mulay*/
/*Roll.No: 02*/
/*Div: A*/

%{
int no_line=0;
int no_space=0;
int no_char=0;
int no_words=0;
#include<string.h>
%}

%%
([a-zA-Z])+ {no_words++; no_char+=strlen(yytext);}
[" "] {no_space++;}
["\n"] {no_line++;}
. ;

%%

int yywrap(){

}

int main(int argc,char* argv[]){
    yyin=fopen("test.txt","r");
    yylex();
    printf("Total Spaces %d\n",no_space);
    printf("Total Words %d\n",no_words);
    printf("Total Line %d\n",no_line);
    no_char+=no_space;
    printf("Total Char %d\n",no_char);
    fclose(yyin);
}
```



```
this is test
this is test
```