Fokane Sakshi Anil
TE-A-42

Page No.
Date

Assignment NO: 7

lex & YACC Program.

- Aim :- Design lex and YCC program to validate type & Syntax of variables declaration in jaua.

- Problem Statement : Write a program using Yacc Specification to implement lexical analysis phase of compiler to validate type & syntax of variable declaration in jaua.

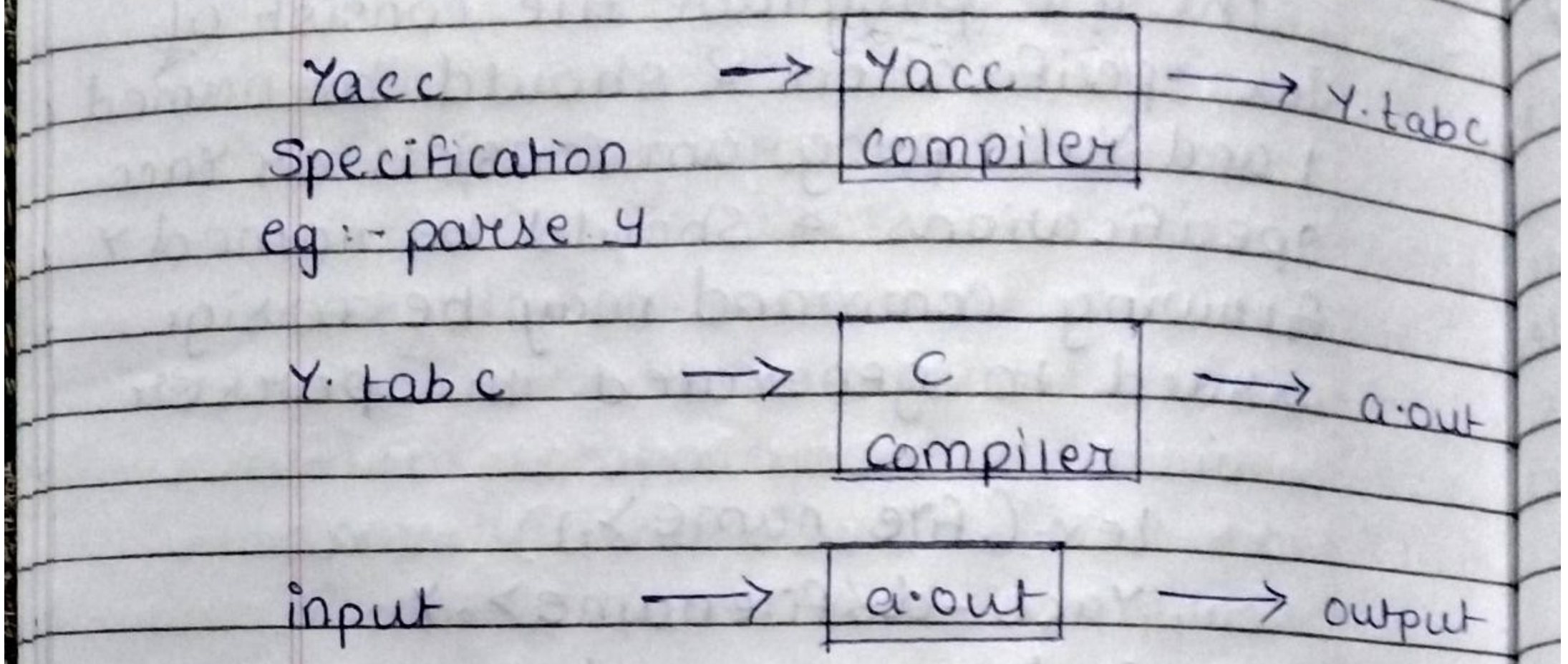
- Pre-requisites :-
lex 110, lex 120, lex 130, lex 140, lex 160, 250

- Software Requirement :-

Sno.	facilities required	Quantity
1	System	1
2	OIS	Ubuntu Kylin
3	S/w name	FLEX YACC.

- Theory -
YACC (Yet Another Compiler Compile) is a Computer program for the UNIX operating System developed by Stephen (Johnson) It is look ahead left to right parser generator generating a parser the part of compiler that tries to make Syntactics sense of source code, specifically a LALR parser based on an analyt

grammer written in notation similar to Backus - Naur form (BNF)



* Structure of yacc file a yacc file looks much like a lex file.

..... definations .....

%. %.

..... rules .....

%. %.

..... code .....

Defination as with lex, all code between %.{ and %.} is copied to the begining of the resulting files rules as with lex a no. of combination of pattern & action. The pattern are now those of context free gramer rather than of regular grammer as was a case with lex code.

* Translating, compiling and Executing a Yacc program:

The lex program file consist of lex specification & should be named l and Yacc program consists or Yacc specifications & should be named Y following command may be ~~assig~~ issued to generated the parser.

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

- Lexical Analyzer for YACC.

The user must supply a lexical Analyzer to read the Stream & Communicate tokens to the parser. The lexical analyzer is an integer valued function called yylex. The relevant portion of the lexical analyzer might look like.

```
yylex(){
    extern int yylval;
    int c;
    ....
    c = getchar();
    .... switch (c) {
    .... case '0';
    ...
    ... case 9;
```

```
yylval = c - '0';
return (DIGIT);
}
```

* Comparing Sentence types:~
1) The Simple Sentence is an independent clause with one Subject & one verb. for example: we are the indian.
2) The Compound Sentence is two or more independent clause joined with Comma, Semicolon & conjunctions.

- Application:
YACC is used to generate parses which are an integral part of compiles

- Conclusion :-
Thus, we have Studied lexical analyzer Syntax & implemented lex & YACC application for Syntax analyzer to validate the given if for expression.

```
//Name: Fokane Sakshi Anil
// Class: TE-A   Rollno: 42
//  ASSINGNMENT:GROUP_B_2

/* Problem Statement  :
Write a program using lex specifications to implement lexical analysis phase of compiler to generate tokens of subset of 'java program'.
*/

/*definition or declaration*/
%{
        #include<stdio.h>
        FILE *fp;
%}

/*Tokenization*/
Package "import".*;
classdef "class".*
inbuiltfun "System.out.println(".*");"
mainfunction "public static void main".*
Assignment [a-zA-Z]+"=".*;
Datatype "int"|"float"|"double"
object .*"=new".*

/*Rules*/
%%
{Package} {printf("Package is %s",yytext);}
{classdef} {printf("Class is %s",yytext);}
{inbuiltfun} {printf("Inbuilt Function is %s",yytext);}
{mainfunction} {printf("Main Function is %s",yytext);}
{Assignment} {printf("Assignment Statement is %s",yytext);}
{Datatype} {printf("Data Type is %s",yytext);}
{object} {printf("%s is object",yytext);}
%%

/*Main Function*/
int main(int argc,char *argv[])
{
        fp=fopen(argv[1],"r");
        yyin=fp;
        yylex();
        return 0;
```

```
}


/* ************JAVA PROGRAM************
//Java input file for lex program
import java.util.Scanner;
class Addition
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		int a,b,sum;
		System.out.println("Enter two numbers : ");
		a=sc.nextInt();
		b=sc.nextInt();

		sum=a+b;
		System.out.println("Sum = "+sum);
	}
}

****************OUTPUT****************
unix@unix-HP-280-G1-MT:~/Desktop/TEB50/Ass.6$ lex lex.l
unix@unix-HP-280-G1-MT:~/Desktop/TEB50/Ass.6$ gcclex.yy.c -ll
unix@unix-HP-280-G1-MT:~/Desktop/TEB50/Ass.6$ ./a.out Addition.java

//Java input file for lex program
Package is import java.util.Scanner;
Class is class Addition
{
	Main Function is public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in); is object
		Data Type is int a,b,sum;
		Inbuilt Function is System.out.println("Enter two numbers : ");
		Assignment Statement is a=sc.nextInt();
		Assignment Statement is b=sc.nextInt();
		Assignment Statement is sum=a+b;
		Inbuilt Function is System.out.println("Sum = "+sum);
	}
}*/
```