

CS3641	Title: Compiler Design Lab	L T P C 0 0 2 1
Version No.	1.0	
Course Prerequisites	Nil	
Objectives	Explain the importance of compiler design. Design and implementation of lexical analyzer using lex tools. Explain the top down and bottom up parsing techniques using programming. Identify the understanding language peculiarities by designing a complete translator for mini language .Explain that computing science theory can be used as the basis for real applications.	
Expected Outcome	Understand the working of lex and yacc compiler for debugging of programs .Understand and define the role of lexical analyzer, use of regular expression and transition diagrams. Understand and use Context free grammar, and parse tree construction. Learn & use the new tools and technologies used for designing a compiler. Develop program for solving parser problems. Learn how to write programs that execute faster.	
List of Experiments		
<ol style="list-style-type: none"> 1. Write a C program to identify whether a given line is a comment or not. 2. Write a C program to recognize strings under 'a','a*b+','abb'. 3. Write a C program to test whether a given identifier is valid or not. 4. Write a C program to simulate lexical analyzer for validating operators. 5. Write a C program for constructing of LL(1) parsing. 6. Write a C program for constructing recursive descent parsing. 7. Write a C program to implement LALR parsing. 8. Write a C program to implement operator precedence parsing. 		
Mode of Evaluation	Internal and External Examinations	
Recommendation by Board of Studies on	14-05-2022	
Date of approval by the Academic Council	20-10-2022	

Requirements: Visual Studio Code/ Dev C++

Experiment No.1 - Write a C program to identify whether a given line is a comment or not

```
#include <stdio.h>
#include <string.h>
#define max 100
void main ()
{
    int a,b;
    char arr[max];
    char dest[2];
    printf("Enter the line\n");
    scanf("%s",arr);
    strcpy(dest,arr,2);
    if ((a=strncmp(dest,"//",2)) && (b=strncmp(dest,"/*",2)))
        printf("not Comment\n");
    else
        printf("Comment\n");
}
```

Performance:

Input Task: //string

Output: comment

Experiment No.2 - Write a C program to recognize strings under 'a','a*b+','abb'

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<string.h>
```

```
#include<stdlib.h>
```

```
void main()
```

```
{
```

```
    char s[20], c;
```

```
    int state=0, i=0;
```

```
    printf("\n Enter a string: ");
```

```
    gets(s);
```

```
    while (s[i]!='0')
```

```
{
```

```
    switch(state)
```

```
{
```

```
    case 0:
```

```
        c = s[i++];
```

```
        if (c=='a')
```

```
            state=1;
```

```
        else if (c=='b')
```

```
            state=2;
```

```
        else
```

```
            state=6;
```

```
        break;
```

```
    case 1:
```

```
        c = s[i++];
```

```
        if (c=='a')
```

```
            state=3;
```

```
        else if (c=='b')
```

```
            state=4;
```

```
        else
```

```
            state=6;
```

```
        break;
```

```
    case 2:
```

```
        c = s[i++];
```

```
        if (c=='a')
```

```
            state=6;
```

```
        else if (c=='b')
```

```
            state=2;
```

```
        else
```

```
            state=6;
```

```
        break;
```

```
    case 3:
```

```
        c = s[i++];
```

```
        if (c=='a')
```

```
            state=3;
```

```
        else if (c=='b')
```

```
            state=2;
```

```
        else
```

```
            state=6;
```

```
        break;
```

```
    case 4:
```

```
        c = s[i++];
```

```
        if (c=='a')
```

```
            state=6;
```

```
        else if (c=='b')
```

```
            state=5;
```

```
        else
```

```
            state=6;
```

```
        break;
```

```
    case 5:
```

```
        c = s[i++];
```

```

        if (c=='a')
            state=6;
        else if (c=='b')
            state=2;
        else
            state=6;
        break;

    case 6:
        printf ("\n %s is not recognised", s);
        exit(0);
    }

if ((state==0) || (state==1) || (state==3))
    printf ("\n %s is accepted under rule 'a*' ", s);
else if ((state==2) || (state==4))
    printf ("\n %s is accepted under rule 'a*b+' ", s);
else if (state==5)
    printf ("\n %s is accepted under rule 'abb' ", s);
getch();
}

```

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
clrscr();
char string[25];
int count=0,flag;
printf("enter any string: ");
gets(string);
if (string[0]>='a'&&string[0]<='z') //small letter
    (string[0]>='A'&&string[0]<='Z') //cap letter
    (string[0]=='_') //underscore
    )

    {
for(int i=1;i<=strlen(string);i++)

{
    if((string[i]>='a'&& string[i]<='z')

    ||
    (string[i]>='A' && string[i]<='Z')

    ||
    (string[i]>='0'&& string[i]<='9')

    ||
    (string[i]=='-')
)

```

```
{  
count++;  
  
}  
}  
  
if(count==strlen(string))  
{  
flag=0;  
}  
}  
  
else  
{  
flag=1;  
}  
  
if(flag==1)  
printf("%s is not valid identifier",string);  
else  
printf("%s is valid identifier",string);  
  
getch();  
}
```