

Fr. Conceicao Rodrigues College of Engineering
Department of Computer Engineering

Academic Term : Jan-May 2024 - 25

Class : T.E. (Computer - A)

Subject Name : System Programming and Compiler Construction

Subject Code : (CPC601)

Practical No:	1
Title:	To write a program for implementing Symbol Table.
Date of Performance:	4/02/2025
Date of Submission:	10/02/2025
Roll No:	9913
Name of the Student:	Mark Lopes

Evaluation:

Sr. No	Rubric	Grade
1	Time Line (2)	
2	Output(3)	
3	Code optimization (2)	
4	Postlab (3)	

Signature of the Teacher :

Experiment No 1

AIM:

To write a program for implementing Symbol Table.

ALGORITHM

Step1: Start the program for performing insert, display, delete, search and modify option in symbol table

Step2: Define the structure of the Symbol Table

Step3: Enter the choice for performing the operations in the symbol Table

Step4: If the entered choice is 1, search the symbol table for the symbol to be inserted. If the symbol is

already present, it displays “Duplicate Symbol”. Else, insert the symbol and the corresponding address in the symbol table.

Step5: If the entered choice is 2, the symbols present in the symbol table are displayed.

Step6: If the entered choice is 3, the symbol to be deleted is searched in the symbol table.

Step7: If it is not found in the symbol table it displays “Label Not found”. Else, the symbol is deleted.

Step8: If the entered choice is 5, the symbol to be modified is searched in the symbol table.

Sample Input and Output:

```
l2sys29@l2sys29-Veriton-M275: ~/Desktop/syedvirus
l2sys29@l2sys29-Veriton-M275:~/Desktop/syedvirus$ ./exp1_syntab
Expression terminated by $:A+B+C=D$
Given Expression:A+B+C=D
Symbol Table
Symbol  addr      type
A       25731088  identifier
+       25731168  operator
B       25731232  identifier
+       25731312  operator
C       25731376  identifier
=       25731456  operator
D       25731536  identifier
l2sys29@l2sys29-Veriton-M275:~/Desktop/syedvirus$
```

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

Department of Computer Engineering

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2024- 25

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<ctype.h>

#define MAX 100

typedef struct{
    char symbol;
    char type[20];
} Symbol;

Symbol symbolTable[MAX];
int symbolCount = 0;

int search(char symbol)
{
    for (int i = 0; i < symbolCount; i++)
    {
        if (symbolTable[i].symbol == symbol){
            return i; // found
        }
    }
    return -1; // not found
}

void insert(char symbol, char *type)
{
    if (search(symbol) != -1)
    {
        printf("Duplicate Symbol: %c \n", symbol);
        return;
    }
    symbolTable[symbolCount].symbol = symbol;
    strcpy(symbolTable[symbolCount].type, type);
    symbolCount++;
}

void createSymbolTable(char *expr)
{
    printf("The expression is: %s \n", expr);

    for (int i = 0; expr[i] != '\0'; i++)
    {
        if (expr[i] == '$')
        {
            break; // expression terminated
        }
        if (isalpha(expr[i])) // If the character is an alphabet (identifier)
        {
```

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

Department of Computer Engineering

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2024- 25

```
        insert(expr[i], "identifier");
    }
    else if (expr[i] == '+' || expr[i] == '-' || expr[i] == '=' || expr[i] == '*') // If it's an operator
    {
        insert(expr[i], "operator");
    }
    else if (isdigit(expr[i])) // If it's a digit (constant)
    {
        insert(expr[i], "constant");
    }
}
printf("Table is created.");
}

void display_table()
{
    printf("\nSymbol Table\n");
    printf("Symbol    Type\n");
    for(int i = 0; i < symbolCount; i++)
    {
        printf("%c    %s\n", symbolTable[i].symbol, symbolTable[i].type);
    }
}

void delete_Symbol(char symbol)
{
    int pos = search(symbol);
    if(pos == -1){
        printf("Symbol not found.");
        return;
    }
    for(int i = pos; i < symbolCount - 1; i++)
    {
        symbolTable[i] = symbolTable[i+1];
    }
    symbolCount--;
    printf("Symbol %c is deleted successfully.\n", symbol);
}

int main()
{
    char expr[MAX];
    int choice;

    while(1) {
        printf("Enter your choice: \n1. Enter your expression\n2. Create Table\n3. Display Table\n4.
Delete Symbol\n5.Exit\n");
        scanf("%d", &choice);

        // Clear the input buffer
        while(getchar() != '\n');
```

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

Department of Computer Engineering

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2024- 25

```
switch (choice)
{
case 1:
    printf("Enter your expression: ");
    fgets(expr, MAX, stdin); // Read the expression

    // Remove newline character if present
    expr[strcspn(expr, "\n")] = 0;
    break;

case 2:
    createSymbolTable(expr);
    break;

case 3:
    display_table();
    break;

case 4:
    char n;
    printf("Enter the symbol to be deleted");
    scanf("%c",&n);
    delete_Symbol(n);
    break;

case 5:
    printf("Exiting program.....");
    return;

default:
    printf("Invalid choice. Please enter a valid option.\n");
    break;
}
}
```

```
Enter your choice:
1. Enter your expression
2. Create Table
3. Display Table
4. Delete Symbol
5.Exit
1
Enter your expression: 12k2+3=5
Enter your choice:
1. Enter your expression
2. Create Table
3. Display Table
4. Delete Symbol
5.Exit
2
The expression is: 12k2+3=5
Duplicate Symbol: 2
Table is created.Enter your choice:
1. Enter your expression
2. Create Table
3. Display Table
4. Delete Symbol
5.Exit
3
```

```
3
Symbol Table
Symbol      Type
1          constant
2          constant
k          identifier
+          operator
3          constant
=          operator
5          constant
Enter your choice:
1. Enter your expression
2. Create Table
3. Display Table
4. Delete Symbol
5.Exit
4
Enter the symbol to be deletedk
Symbol k is deleted successfully.
```

```
Enter your choice:
1. Enter your expression
2. Create Table
3. Display Table
4. Delete Symbol
5.Exit
3
```

```
Symbol Table
Symbol      Type
1          constant
2          constant
+          operator
3          constant
=          operator
5          constant
```

```
Enter your choice:
1. Enter your expression
2. Create Table
3. Display Table
4. Delete Symbol
5.Exit
5
```

```
Exiting program.....
```

```
D:\C++\Users\Mark Lopez\Desktop\college\Sem 6\snpc>
```


Postlab Questions:

1. Explain different phases of compiler. Illustrate all the output after each phase for the following statement

$a = b + c - d * 5$

SPCC

Compiler construction (C.C)

Symbol table

error handling

1. Lexical analysis

gives from left to right, classifies them according to categories that can be implemented using regex and give tokens

2. Syntax analysis

recognizes all spaces, tabs and checks for grammar and gives parse trees

3. Semantic analysis

checks for if left and right side is of same type.

ex: $\text{int } y = 18;$ gives syntax tree

4. Intermediate code generation

checks whether it is in form of '3 address code' (it should fit in 3 address location) if not, then convert in 3 address code.

ex:

$$a = b + c + d * e$$

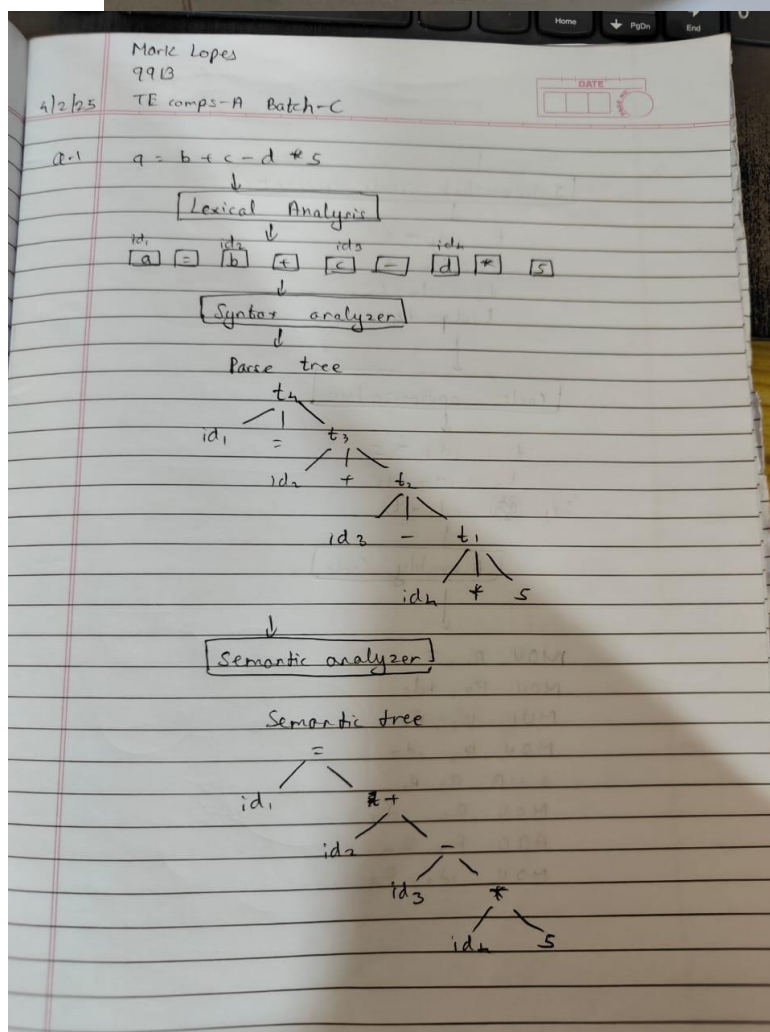
Tree diagram for $a = b + c + d * e$:

```

    graph TD
      A["1 2 3 4 5  
a = b + c + d * e"]
      A --> B["t1 = d * e"]
      A --> C["t2 = c + t1"]
      A --> D["a = b + t2"]
  
```

5. Code optimization :-

Make code more optimized by removing redundancies in code, i.e. it will less time to compile/execute.



↓
Intermediate code generator

↓
 $t_1 = id_4 * 5$
 $t_2 = id_3 - t_1$
 $t_3 = id_2 + t_2$
 $id_1 = t_3$

↓
code optimization

↓
 $t_1 = id_4 * 5$
 $t_2 = id_3 - t_1$
 $id_1 = id_2 + t_2$

↓
Assembly Code

↓
MOV R₁, 5
MOV R₂, id₃
MUL R₁, R₂
MOV R₃, id₂
SUB R₃, R₁
MOV R₄, id₂
ADD R₃, R₄
MOV id₁, R₃