Practical: 14

Aim: Identify the constructs and implement the symbol table for the chosen language.

Program:

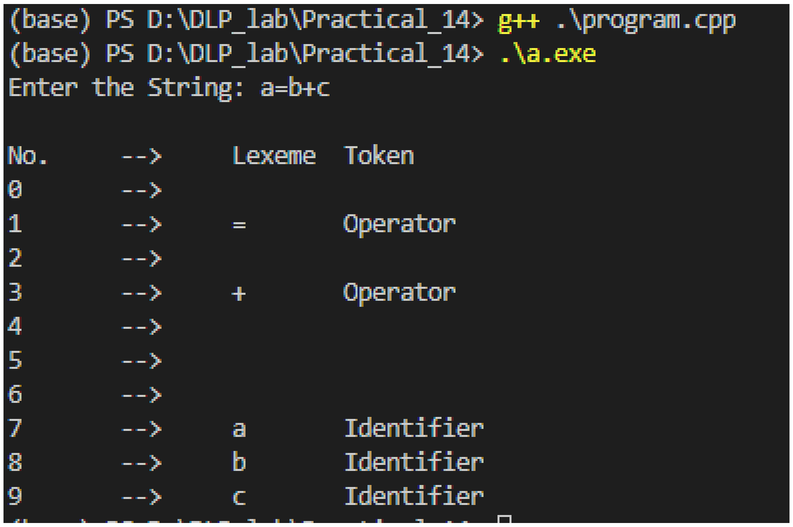
```
/*
    author: mr_bhishm
    created: 17-10-2020 22:07:54
    "Make it work, make it right, make it fast."
                                                – Kent Beck
*/
#include<stdio.h>
#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

char hash1[10]= {};
string token[10]= {};
void hash2(string in){
    for(int i=0; i<in.length(); i++){
        int a = in[i];
        int mod;
        mod = a % 10;
        if(hash1[mod]==0){
            hash1[mod] = in[i];
            if(isalpha(in[i])){
                token[mod] = "Identifier";
            }else{
                token[mod] = "Operator";
            }
        }else{
            mod++;
            while(hash1[mod]!=0){
                mod++;
                if(mod>9){
                    mod=0;
                }
            }
            hash1[mod] = in[i];
            if(isalpha(in[i])){
                token[mod] = "identifier";
            }
            else{
                token[mod] = "operator";
            }
        }
    }
}
```

```
int main(){
    string input;
    cout<<"Enter the String: ";
    cin>>input;

    hash2(input);
    cout<<"\nNo.\t-->\tLexeme\tToken";
    for(int j=0 ; j<10; j++){
        cout<<endl<<j<<"\t-->\t"<<hash1[j]<<"\t"<<token[j];
    }
    return 0;
}
```

Output:



```
(base) PS D:\DLP_lab\Practical_14> g++ .\program.cpp
(base) PS D:\DLP_lab\Practical_14> .\a.exe
Enter the String: a=b+c

No.     -->     Lexeme  Token
0       -->
1       -->     =       Operator
2       -->
3       -->     +       Operator
4       -->
5       -->
6       -->
7       -->     a       Identifier
8       -->     b       Identifier
9       -->     c       Identifier
```

Conclusion: From this practical I have learnt about how to implement symbol table for any chosen language.