Experiment No. 4

Date:

Aim: Write a C++ program to implement the concept of local variable & Global variable.

Experiment No. 4

Date :

Aim : Write a C++ program to implement the concept of local variable & global variable.

Theory : In C++, local variables are variables that are declared inside a function, block or method. They are also known as 'automatic' variables beoz their lifetime is automatically managed by the system.

Local variables can be also declared with the static keyword, which extends their lifetime to the entire duration of the program but keeps their scope local to the function.

Code :

```
#include<iostream>
void examplefunction(){
    int localVar = 10;
    Std::cout<<"Local variable inside the function
    is:"<<localVar<<endl;
}
int main(){
    examplefunction();
// local var is not accessible here as it is local
// to the examplefunction.
return 0;
}
```

output: Local variable inside the function : 10.

Theory: Global variables in C++ are the variables that are declared outside of all functions, typically at the top of a program. These variables can be accessed from any part of the program, including from within functions, classes & blocks.
Global variables are a useful tool in some cases, but they should be used carefully to avoid issues with maintainibility, readability & debugging.

Code:

```
#include<iostream>
using namespace std;

int globalVar = 5; // Global variable

void ModifyGlobalVar (){
    globalVar = 10; // modified global variable
    cout << "Global variable inside function
            modifyGlobalVar () : " << globalVar <<
            endl;
}
```

Conclusion: This practical helped highlighting the importance of variable scope management in controlling the data flow & maintaining program structure.

```
int main () {
    cout << "Global variable in the main before
            modification :" << globalVar << endl;
    ModifyGlobalVar (); // call the function to
    modify the global variable.
    cout << "Global variable in the main after
            modification :" << endl;
    return 0;
}
```

Output :

```
Global variable in the main before modification
:5
Global variable inside function ModifyGlobalVariable
():10
Global variable in main after modification
:10
```

Conclusion: This practical helped highlighting the importance of variable scope management in controlling the data flow and maintaining program structure.