Date __/__/__ Experiment No 7

Saathi

Title :- Program to implement call by reference and return by value concept

Aim :- To write a program to implement call by reference and return by value concept.

Theory :-

Call By Reference :-

The call by reference method of passing arguments to a function copies the reference of an argument into the formal parameter. Inside the function, the reference is used to access the actual argument used in the call. This means that changes made to the parameter affect the passed argument.

To pass the value reference, argument reference is passed to the function just like any other value. So accordingly you need to declare the function parameters as reference types as in the following function swap(); which exchanges the value of the two integer pointed to by its arguments.

Date __/__/____

Return by value :-

Return by value is the simplest and safest return type to use. When a value is returned by value, a copy of that value is returned to the caller. As with pass by value, you can return by value literals, variables (ex. x), or expressions, which makes return by value very flexible.

Another advantage of return by value is that you can return variables (or expression) that involves local declared within the function without having to worry about scoping issues. Because the variable are evaluted before the functions returns, and a copy of the value is returned to the caller, there are no problems when the functions variable goes out of scope at the end of the function.

PROGRAM :-

```
#include <iostream>  // header file
using namespace std;

int product_returnvalue (int a , int b)
  {     return  a*b;
  }
```

```
void swap_callbyreturnvalue (int &a, int &b)
{
    int tmp = a;
    a = b;
    b = tmp;
}
int main()                    //main function.
{ int a=5, b=12;

cout<<"values before swaping: "<<endl;
cout<<"A: "<<a<<endl;
cout<<"B: "<<b<<endl;

swap_callbyreturnvalue (a,b);
cout<<"value after swaping: "<<endl;
cout<<"A: "<<a<<endl
    <<"B" <<b<<endl;

cout<<"product of a & b :" <<
    product_returnvalue (a,b) <<endl;

    return 0;

}
```

Date __/__/____

Output :

Values before swapping :

A = 5

B = 12

Values after swaping

A = 12

B = 15

Product of a & b : 60

Conclusion :-

By doing this practical, I understood the concept of call by reference & return by value.