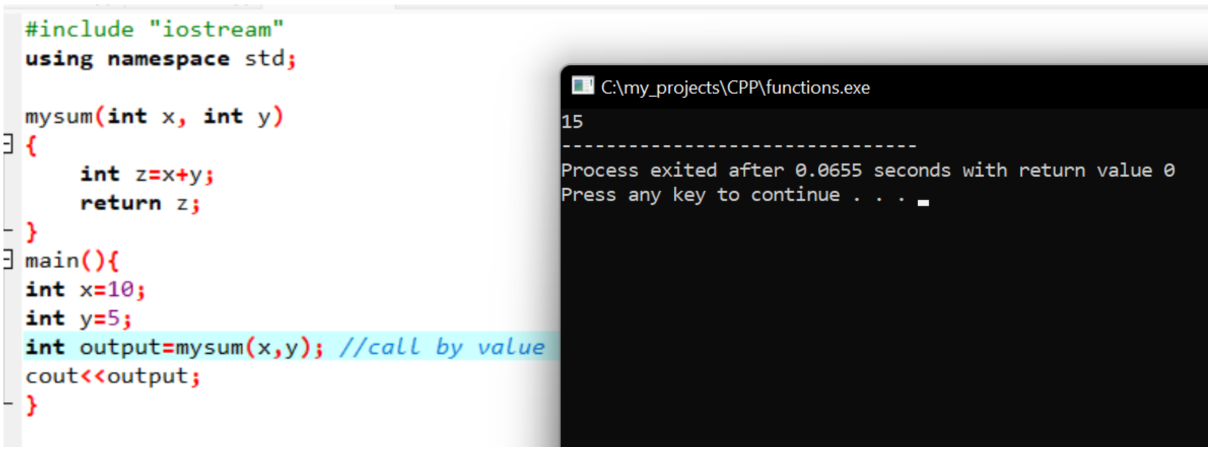
Summary CPP

Sessions No 10(02-11-2022)

- **Functions** get space for their variables from Stack memory, So whenever a function is called one ***Activation Record*** of that function is being created in **Stack memory**. As we know ***main()*** is the entry point for our CPP code, So whenever we run a CPP program automatically an Activation record for main() gets created in Stack memory. Now as different functions have their separate activation record, So if we create a variable with the same name in two functions simultaneously then it will not create any issue, as they are been declared in their private space.
  Variables defined inside functions is a **local variables**.
  As the "**return**" keyword comes, the function returns the data, and the Activation record of that function vanishes out from Stack memory.

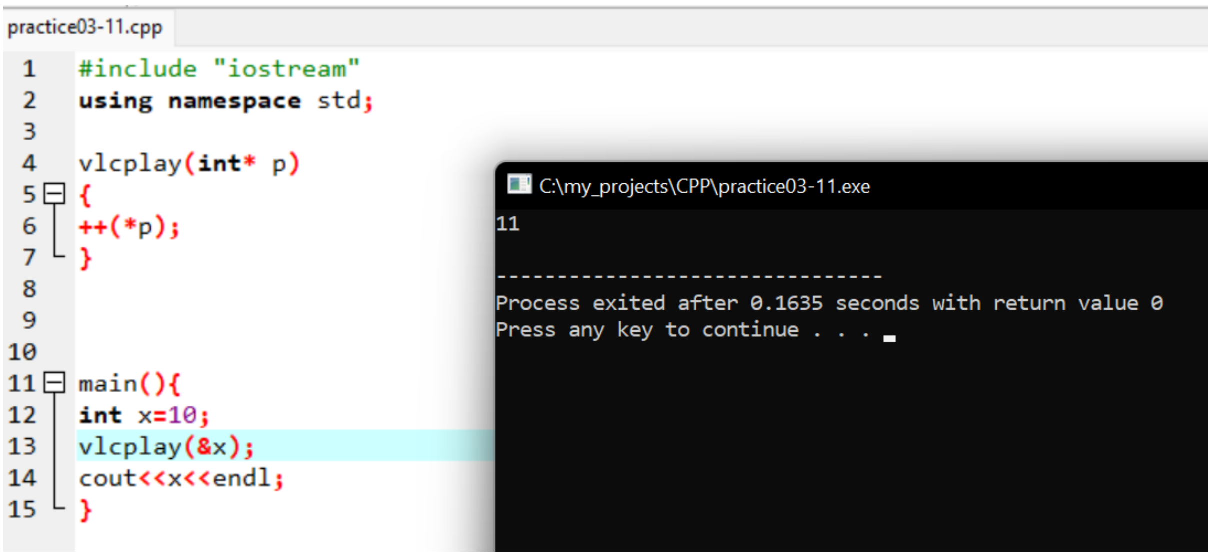
- **Call by value** - When we create a function and while calling that function we send it some **value**(as an argument) then it is known as we are calling the function by value. Now in the given example, we have created a function named "*mysum*" which is taking two values from the main function. In this case, We have variables x and y in both functions, but they are working separately. Because they are in different activation records.



```
#include "iostream"
using namespace std;

mysum(int x, int y)
{
     int z=x+y;
     return z;
}
main(){
int x=10;
int y=5;
int output=mysum(x,y); //call by value
cout<<output;
}
```

```
C:\my_projects\CPP\functions.exe
15
--------------------------------
Process exited after 0.0655 seconds with return value 0
Press any key to continue . . .
```

- **Call by address** - Here we call a function and provide the address of a variable to it as an argument, Now this address can be received by a function, And it can store it in a pointer variable and use it.



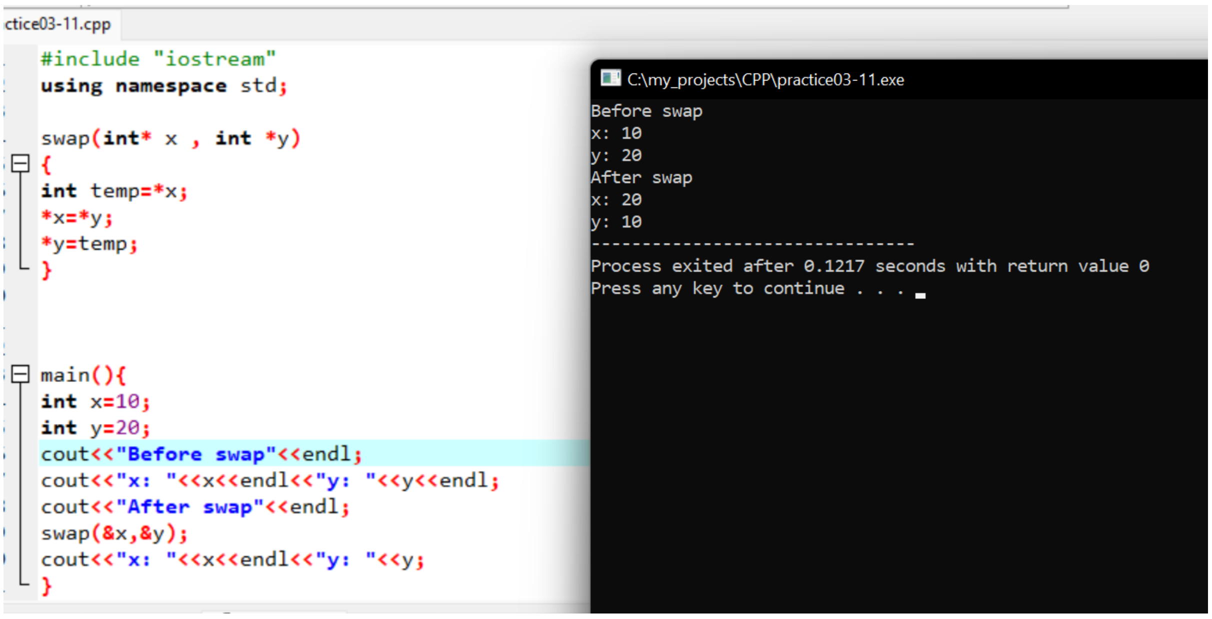
practice03-11.cpp

```cpp
#include "iostream"
using namespace std;

vlcplay(int* p)
{
++(*p);
}



main(){
int x=10;
vlcplay(&x);
cout<<x<<endl;
}
```

C:\my_projects\CPP\practice03-11.exe

```
11

--------------------------------
Process exited after 0.1635 seconds with return value 0
Press any key to continue . . .
```

- **Swap The numbers -** Here we have created a swap function that will swap the value of our variables. Here we are passing the address of variables to the function. Now the function is receiving both addresses in the pointers, And then handling both variables' data with the help of pointers.



ctice03-11.cpp

```cpp
#include "iostream"
using namespace std;

swap(int* x , int *y)
{
int temp=*x;
*x=*y;
*y=temp;
}


main(){
int x=10;
int y=20;
cout<<"Before swap"<<endl;
cout<<"x: "<<x<<endl<<"y: "<<y<<endl;
cout<<"After swap"<<endl;
swap(&x,&y);
cout<<"x: "<<x<<endl<<"y: "<<y;
}
```

C:\my_projects\CPP\practice03-11.exe

```
Before swap
x: 10
y: 20
After swap
x: 20
y: 10
--------------------------------
Process exited after 0.1217 seconds with return value 0
Press any key to continue . . .
```