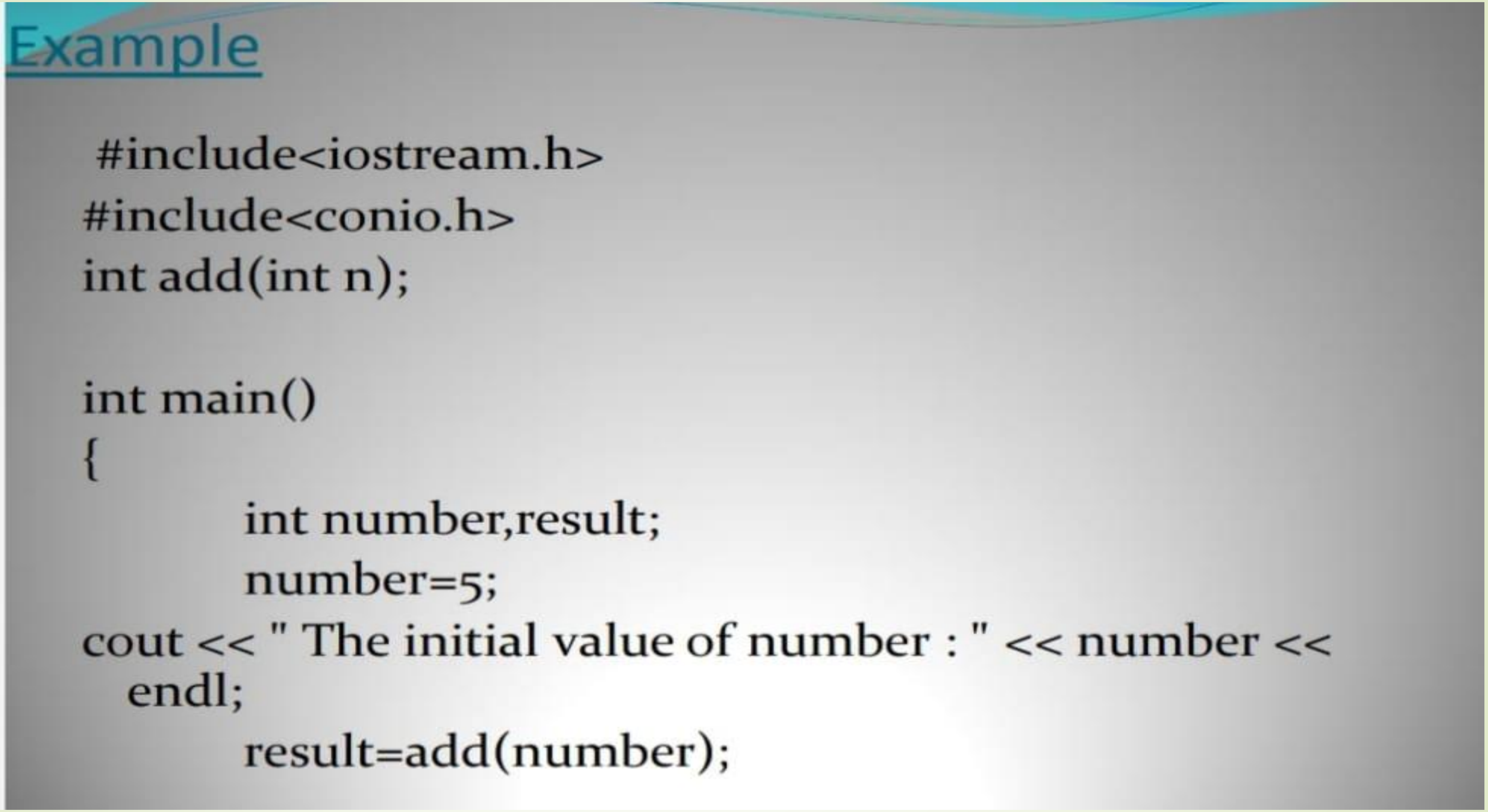
FUNCTION

PRESENTED BY – ANJALI SONA

DEPARTMENT OF COMPUTER SCIENCE

Pass/Call by value

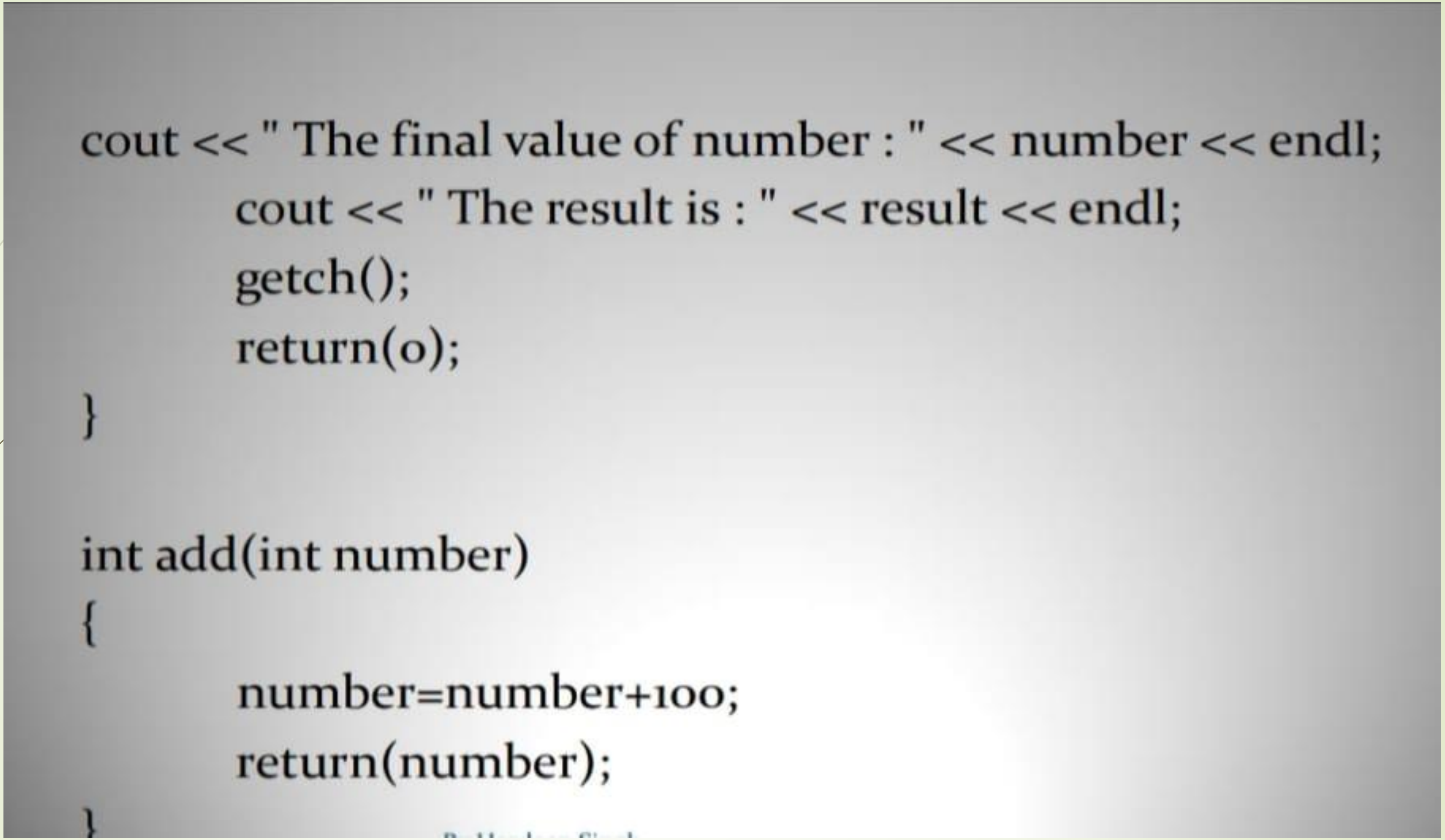
- Copies of the arguments are created .
- The parameters are mapped to the copies of the arguments created.
- The changes made to the parameter do not affect the arguments.



Example

```
 #include<iostream.h>
#include<conio.h>
int add(int n);

int main()
{
           int number,result;
           number=5;
cout << " The initial value of number : " << number <<
   endl;
           result=add(number);
```

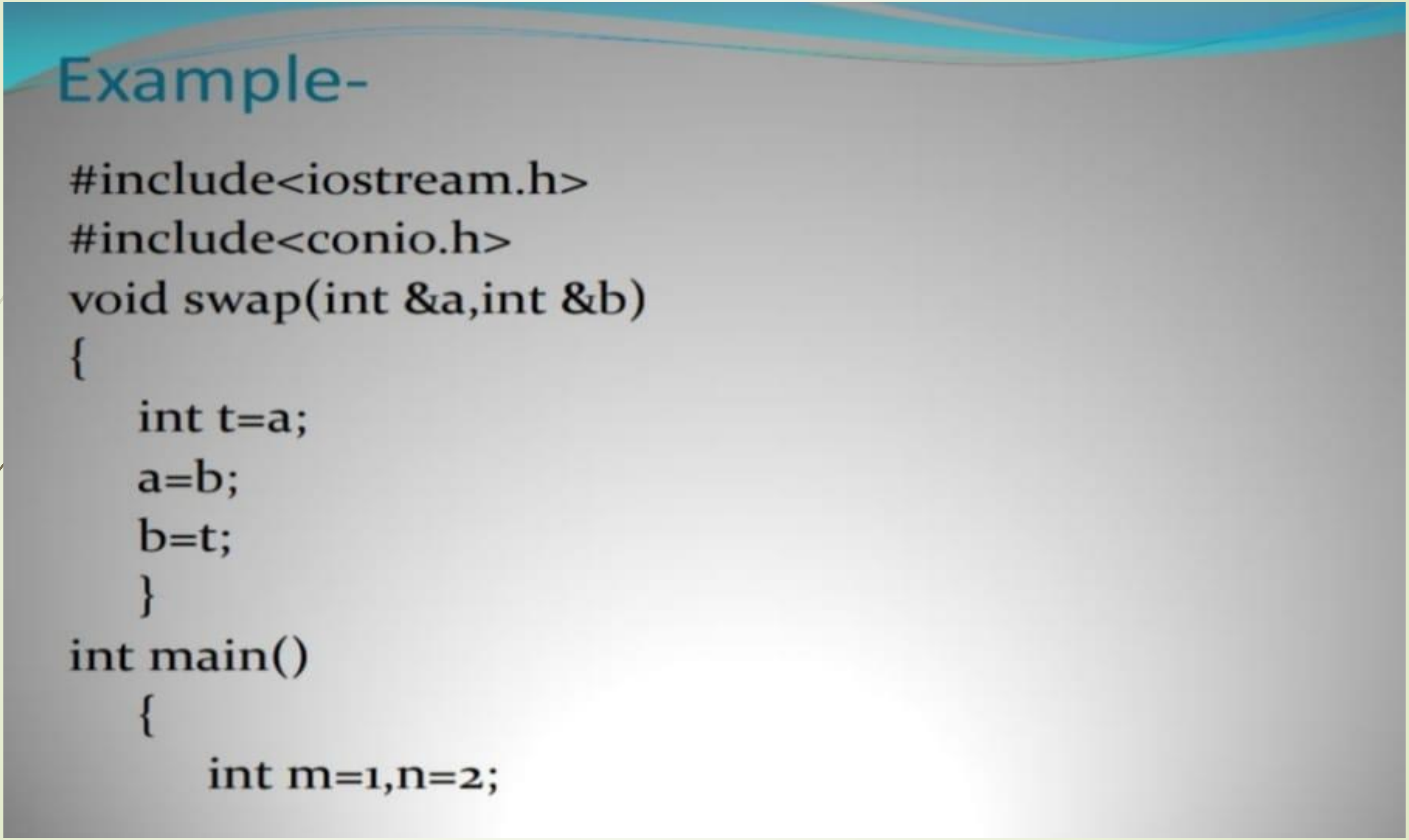


```
cout << " The final value of number : " << number << endl;
        cout << " The result is : " << result << endl;
        getch();
        return(0);
}

int add(int number)
{
        number=number+100;
        return(number);
}
```

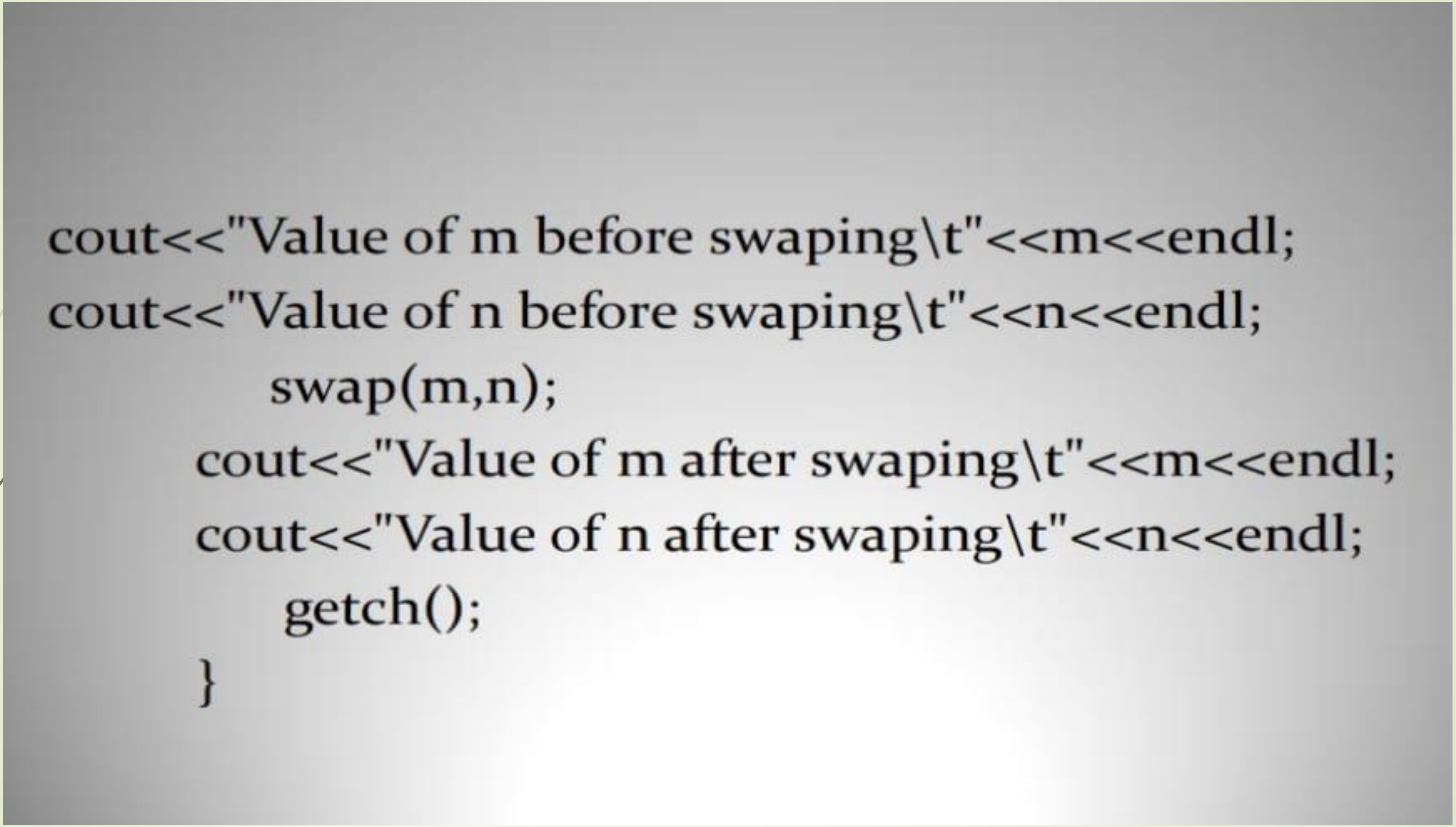
Pass/Call by reference

- Pass by reference is the second way of passing parameters to the function.
- The address of the argument is copied into the parameter.
- The changes made to the parameter affect the arguments.



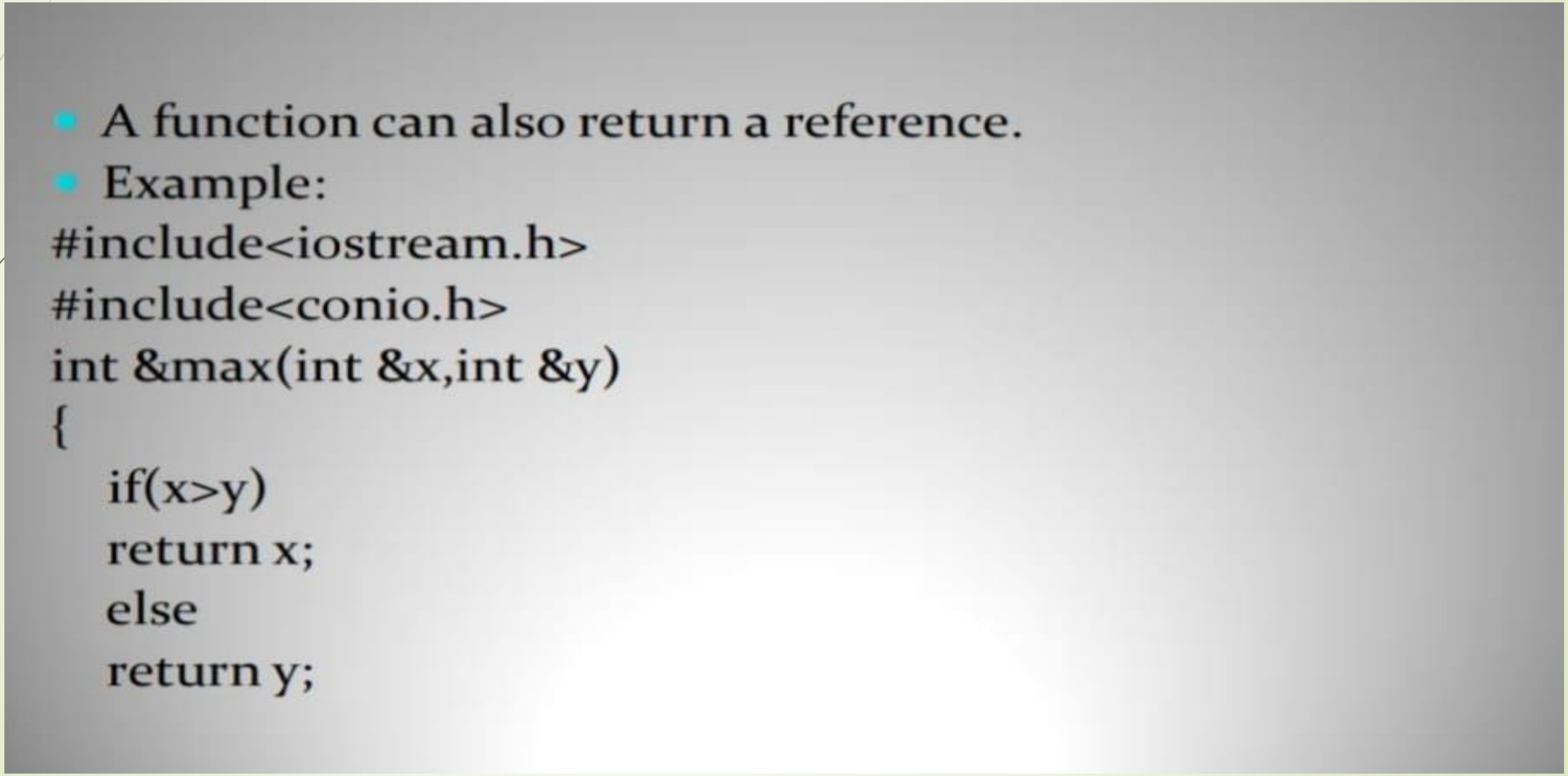
Example-

```
#include<iostream.h>
#include<conio.h>
void swap(int &a,int &b)
{
    int t=a;
    a=b;
    b=t;
    }
int main()
    {
        int m=1,n=2;
```



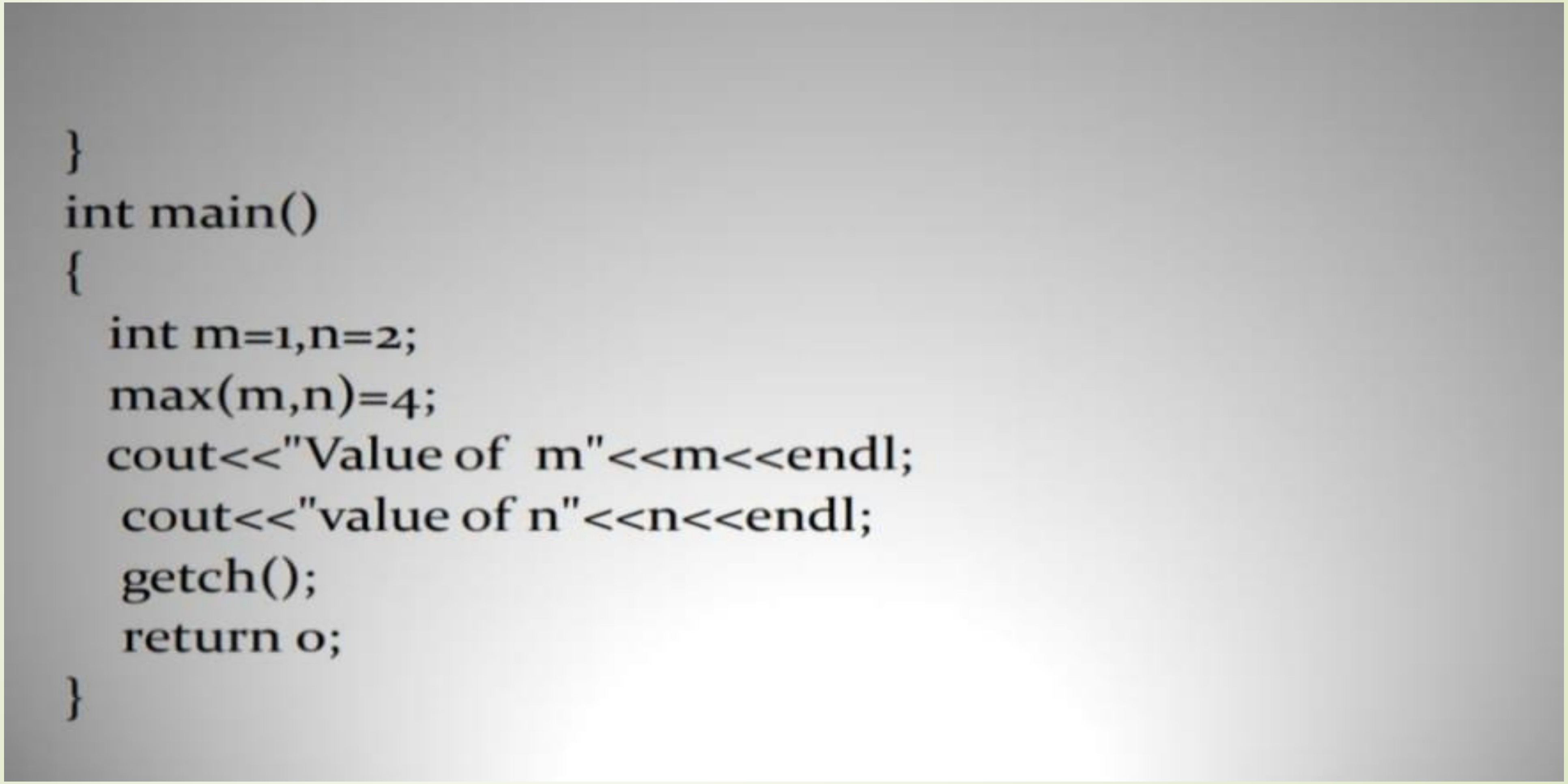
```
cout<<"Value of m before swaping\t"<<m<<endl;
cout<<"Value of n before swaping\t"<<n<<endl;
        swap(m,n);
     cout<<"Value of m after swaping\t"<<m<<endl;
     cout<<"Value of n after swaping\t"<<n<<endl;
         getch();
     }
```

Return by reference



- A function can also return a reference.
- Example:

```
#include<iostream.h>
#include<conio.h>
int &max(int &x,int &y)
{
   if(x>y)
   return x;
   else
   return y;
```



```
}
int main()
{
   int m=1,n=2;
   max(m,n)=4;
   cout<<"Value of  m"<<m<<endl;
    cout<<"value of n"<<n<<endl;
    getch();
    return 0;
}
```

Inline Functions

□An inline function is a function that expanded in line when it is invoked.

□That is the compiler replaces the function call with the corresponding function code .

□ Syntax:

```
 inline function-header
{
Function body
}
```

Default Arguments

- Default values are specified when the function is declared.

- Compier looks at the prototype to see how many arguments function uses.

- Default arguments are useful in situations where some arguments always have the same value.

Function Overloading

□A function is overloaded when same name is given to different function.

□The two functions with the same name will differ at least in one of the following.

a) The number of parameters

b) The data type of parameters

c) The order of appearance

THANK YOU