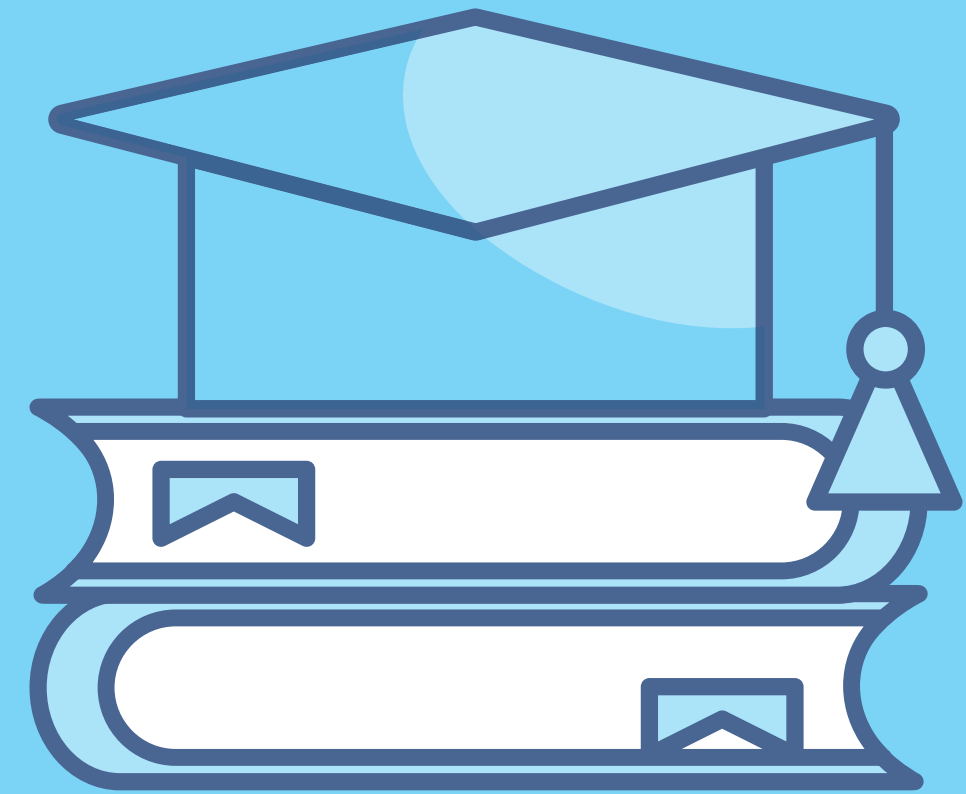




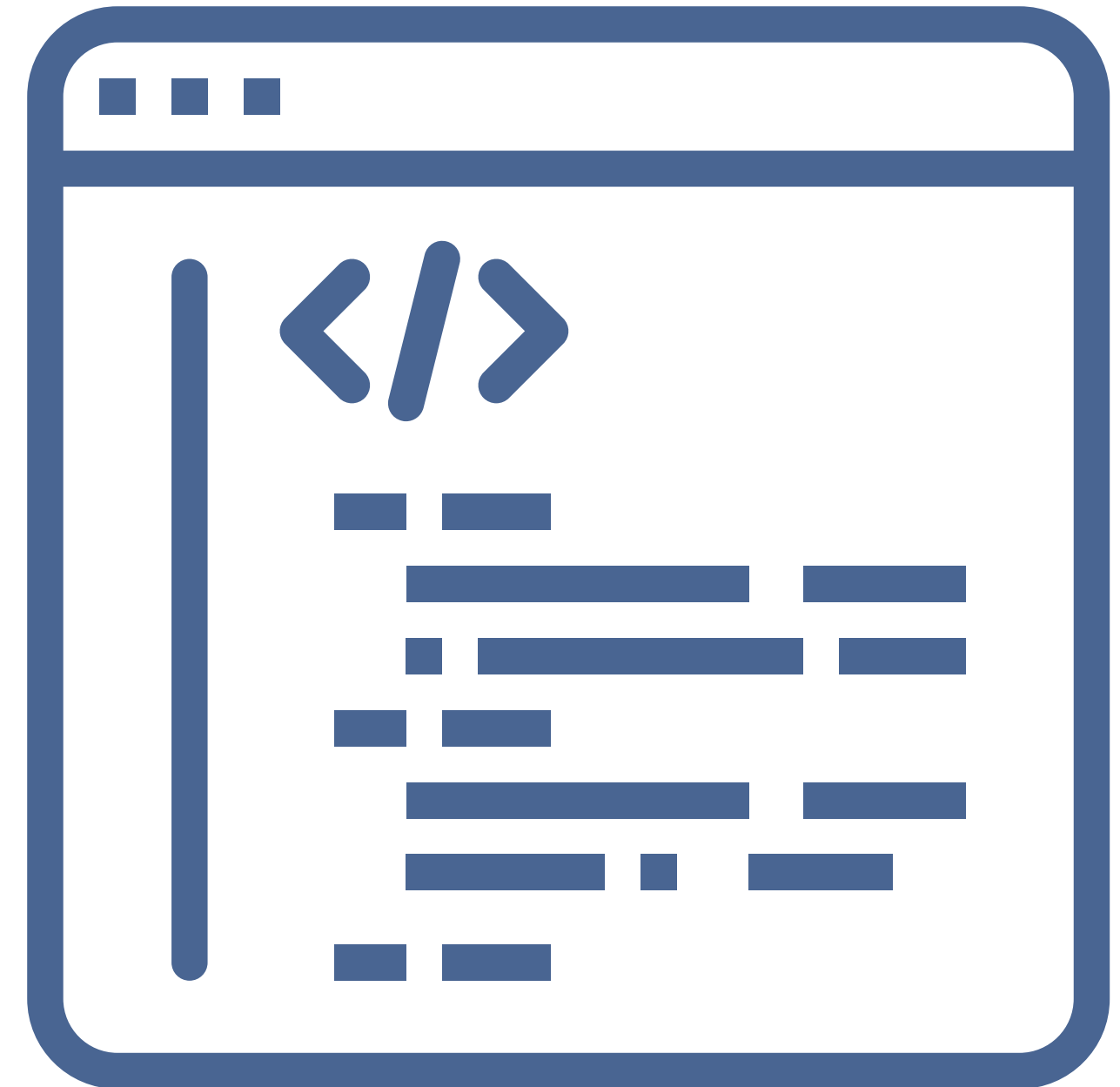
FUNCTION CALL BY VALUE

OBJECT ORIENTED PROGRAMMING (OE-EC604C)
SOUVIK GHOSH 13000320025
ELECTRONICS & COMMUNICATION ENGINEERING



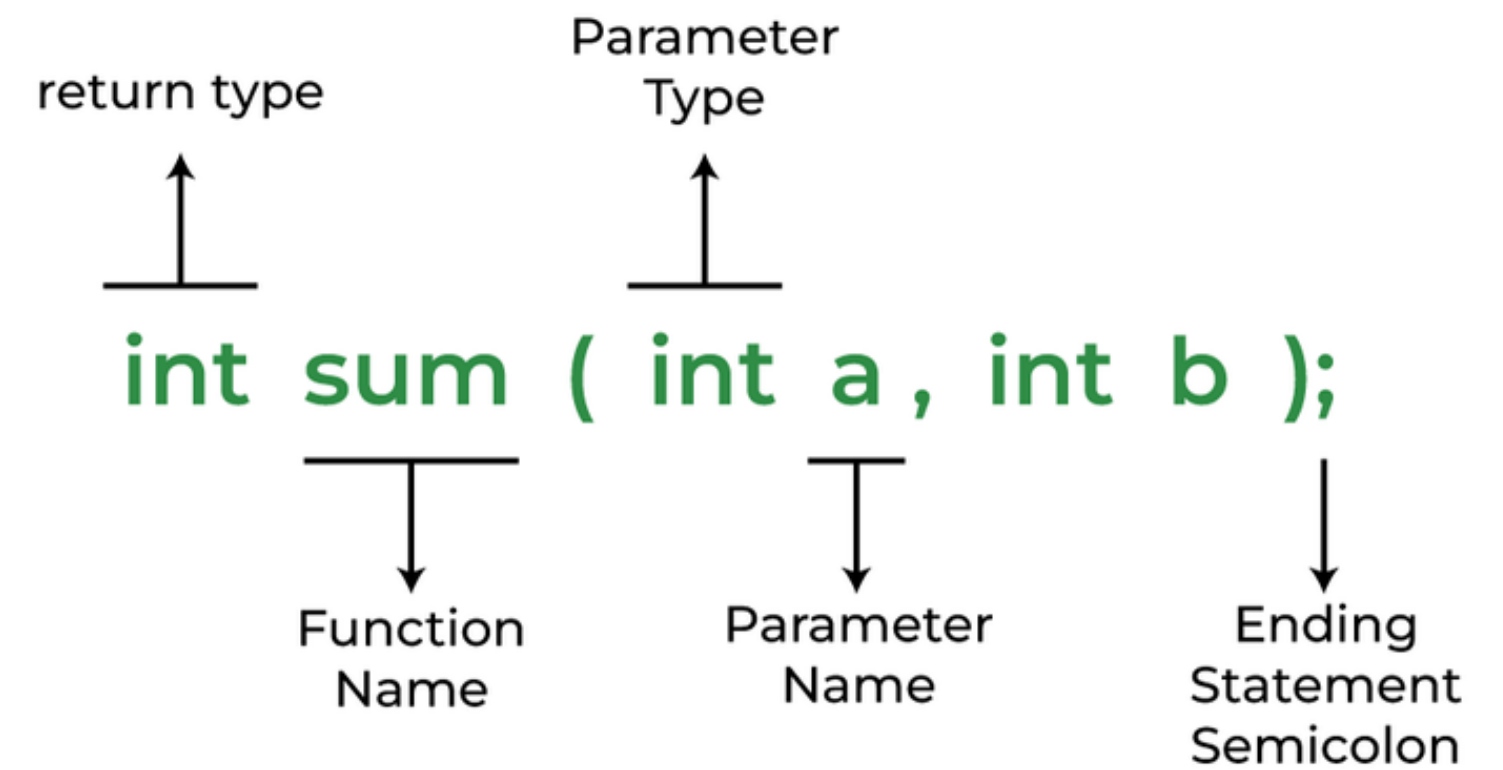
Content

-  **Introduction** 
-  **Types** 
-  **Call by Value and It's Uses** 
-  **Limitations** 
-  **Example** 



Introduction

- A function is a collection of statements that accept inputs, carry out certain calculations, and output the results.
- The concept is to group similar or often performed actions into a function.
- A function is a section of code that executes only when it is called.



Function in C++



Types

In a programming language, one can invoke functions in two ways:



Call by Value

The call-by-value method allows us to copy the actual parameter to a formal parameter. In this case, if we change the formal parameter then the actual parameter doesn't change. C++ uses call by value method by default.

Use of Call by value

Call by value is used in certain conditions like:

- When we do not want to change the actual parameters of the function.
- When we want to make copies of the data instead of the actual data.
- When space is not an issue.



Limitations of using Call by Value

Although call by value is so useful for us while programming in C++, it has a few limitations too:

- Data passed is then stored in temporary memory.
- Can't operate over the actual data rather need to work on a temporary copy made of the data, because of which changes made in the data in the function are not reflected in the main function.
- Memory Space required while using string, array, vector, etc can be huge.
- Tackling Backtracking and recursion can be complex using call-by values.



Example

In the example, if we want to change the value of a variable through a function using call by value then it is only changed in that function only and the actual value of the variable where it is defined is not changed. But, if we want to use the value from the function return the value to the main function.

```
1 // C++ Program to demonstrate Call by value
2 #include <iostream>
3 using namespace std;
4 // Function to change value of x
5 int changeValue(int x)
6 {
7     x += 1;
8     cout<<"Value of X after modifying in function: "<< x << endl;
9 }
10 // main function
11 int main()
12 {
13     int x = 4;
14     cout << "Value of X before modifying: " << x << endl;
15     changeValue(x);
16     cout << "Value of X after modifying in main: " << x << endl;
17     return 0;
18 }
19
```

Value of X before modifying: 4
Value of X after modifying in function: 5
Value of X after modifying in main: 4





Thank
you!