**C++ 101 – Session 9 Notes**

**Topic: Functions in C++**

**1. What is a Function?**

A **function** is a reusable block of code that performs a specific task.  
Instead of repeating code, you define it once in a function and **call it whenever needed**.

**🧱 A Function has 5 main parts:**

| **Part** | **Example** | **Description** |
| --- | --- | --- |
| **Return type** | int, void, string | What the function gives back (if anything) |
| **Function name** | sum, print\_my\_name | Name used to call the function |
| **Parameters** | (int a, int b) | Input values the function uses |
| **Function body** | { ... } | The block of code that runs when the function is called |
| **Return value** | return a + b; | The value sent back to the caller (if not void) |

**2. Defining a Function**

**🔹 Syntax:**

returnType functionName(parameters) {

// block of code

return value; // optional (depends on returnType)

}

**Example 1: A function that prints a message**

void myFunction() {

cout << "I just got executed!" << endl;

}

* void → no return value
* myFunction → function name
* No parameters
* Just prints a message

**Example 2: A function that returns a sum**

int sum(int a, int b) {

return a + b;

}

* Takes two integers
* Adds them
* Returns the result

**Example 3: A function that returns a string**

cpp

CopyEdit

string print\_my\_name(string name) {

return "My name is " + name;

}

**3. Calling (Invoking) a Function**

Once a function is defined, you **call** it by using its name and passing the required arguments (if any).

**✅ Example:**

myFunction(); // calls the void function

cout << sum(5, 10); // calls sum() and prints the result

string output = print\_my\_name("Ngambo");

cout << output;

🧠 **Note:** You can also assign the return value to a variable and use it later.

**4. The Class Code Breakdown**

#include <iostream>

using namespace std;

// A void function (doesn't return anything)

void myFunction() {

cout << "I just got executed!" << endl;

}

// Function that returns the sum of two integers

int sum(int a, int b) {

return a + b;

}

// Function that returns a string with a name

string print\_my\_name(string name) {

return "My name is " + name;

}

int main() {

int x, y;

myFunction(); // Prints message

cout << print\_my\_name("Ngambo") << endl;

cout << "Enter two integers: " << endl;

cin >> x >> y;

cout << "The sum of x and y is: " << sum(x, y) << endl;

return 0;

}

**🧾 Sample Output:**

I just got executed!

My name is Ngambo

Enter two integers:

4

6

The sum of x and y is: 10

**✅ 5. Key Concepts Covered**

| **Concept** | **Example** | **Explanation** |
| --- | --- | --- |
| Declaring a function | int sum(int a, int b) | Introduces the function |
| Calling a function | sum(2, 3) | Executes the function |
| Using void | void myFunction() | No return value |
| Returning a value | return a + b; | Sends result back to caller |
| Assigning return value to a var | int total = sum(x, y); | You can store and use the result |
| Printing directly | cout << sum(x, y); | You can output directly |

**⚠️ 6. Not Covered Yet (Coming Soon)**

We’ve only covered basic function usage. We’ll soon look into:

* 🔄 Function **overloading**
* 🔁 **Recursion**
* 🌐 Function **scope** and variable lifetime
* 🧩 **Default parameters**

**🛠️ Task**

Write a program that:

1. Defines a function to multiply two numbers
2. Defines a function that returns a greeting with your name
3. Calls both functions from main
4. Prints the results

📌 Bonus: Try to pass user input to both functions