Tutorial 15 - Functions & Function Prototypes in C++

Functions in C++

- Functions break code into reusable pieces, promoting modularity and clarity.
- Syntax Example:

```
int sum(int a, int b) {
    int c = a + b;
    return c;
}

```

- **Key Points:**
  - `sum` is a function that takes two integers (`a` and `b`), adds them, and returns the result (`c`).
  - Functions can accept parameters (inputs) and return values.

Calling a Function

- Example:

```
int main() {
    int num1, num2;
    cout << "Enter first number: ";
    cin >> num1;
    cout << "Enter second number: ";
    cin >> num2;
    cout << "The sum is " << sum(num1, num2);
    return 0;
}

```

- **Key Points:**
  - `num1` and `num2` are **actual parameters** passed to the function `sum`.
  - The function performs addition and returns the result.

Function Prototypes

- A **function prototype** declares a function before its definition, helping the compiler recognize it.
- Syntax:

```
int sum(int a, int b);  // Acceptable
int sum(int, int);      // Acceptable
int sum(int a, b);      // Not Acceptable

```

- **Key Points:**
  - Prototype specifies the function name, return type, and parameter types.
  - Parameters can be named or unnamed in prototypes.

Types of Parameters

- **Formal Parameters:** Declared in the function definition (e.g., `int a, int b` in `sum`).
- **Actual Parameters:** Passed during function calls (e.g., `num1, num2` in `sum(num1, num2)`).

Void Functions

- Functions without a return value or parameters:

```
void g() {
    cout << "Hello, Good Morning";
}

```

- **Key Points:**
  - `void` means the function doesn't return anything.
  - Example call: `g();`

Complete Example Code

```
#include<iostream>
using namespace std;
// Function Prototypes
int sum(int, int);
void g();
int main() {
    int num1, num2;
    cout << "Enter first number: ";
    cin >> num1;
    cout << "Enter second number: ";
    cin >> num2;
    cout << "The sum is " << sum(num1, num2);
    g();
    return 0;
}
int sum(int a, int b) {
    return a + b;
}
void g() {
    cout << "\nHello, Good Morning";
}
```