Void VS Value Returning Functions

Review: Working Example

Write a C++ program that inputs two numbers from the user and prints the sum of those two numbers by calling the sum function.

Review

```
#include <iostream>
using namespace std;

int addition(int num1, int num2);

main(){
    float number1, number2, result;
    cout << "Enter First Number: ";
    cin >> number1;
    cout << "Enter Second Number: ";
    cin >> number2;
    result = addition(number1, number2);
    cout << "Sum is: " << result;
}

int addition(int num1, int num2)
{
    int sum = num1 + num2;
    return sum;
}
```

Function Prototype

Function Call

Function Definition

Review

```
#include <iostream>
using namespace std;

int addition(int num1, int num2);

main(){
    float number1, number2, result;
    cout << "Enter First Number: ";
    cin >> number1;
    cout << "Enter Second Number: ";
    cin >> number2;
    result = addition(number1, number2);
    cout << "Sum is: " << result;
}

int addition(int num1, int num2)
{
    int sum = num1 + num2;
    return sum;
}
```

Function Prototype

Function Call

2 Parameters

Value returning Function

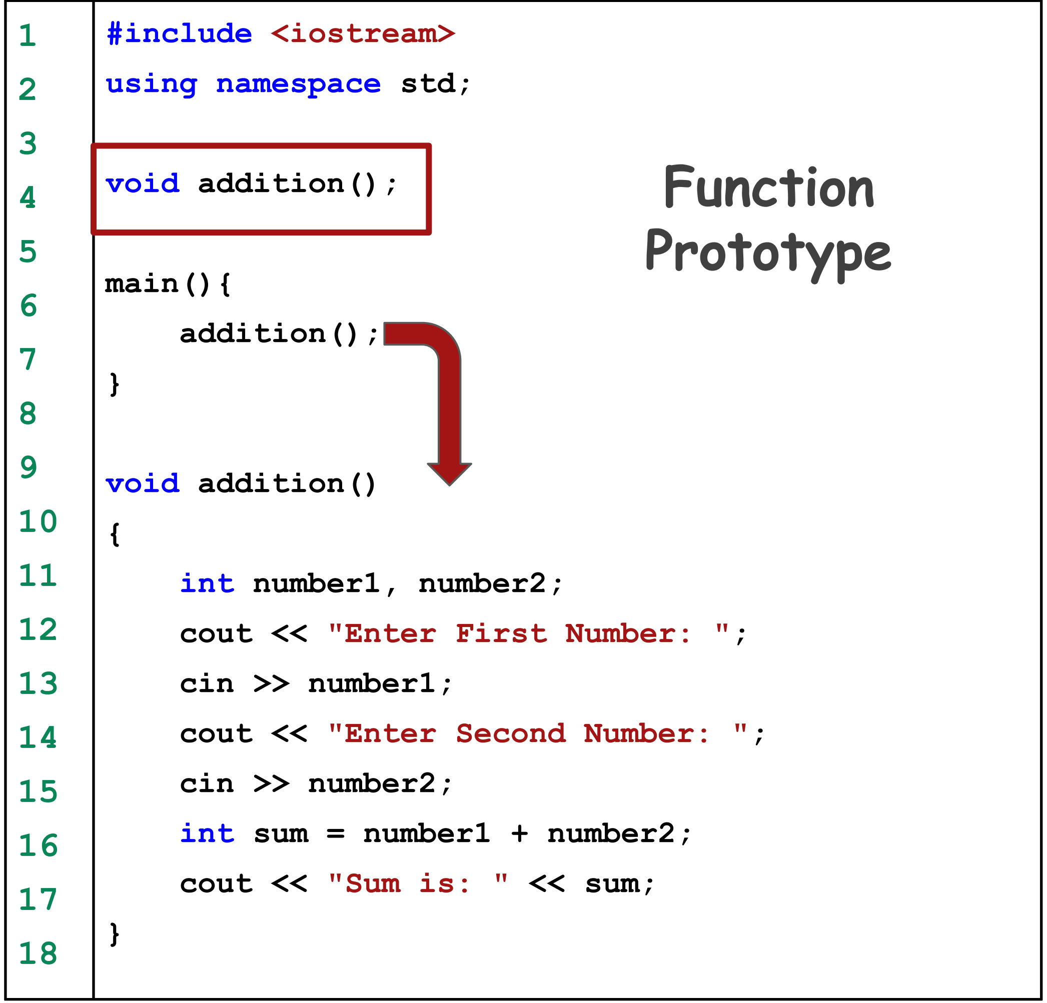
Review

We can also write a function that **doesn't take any input** and **doesn't return any output.**

```
#include <iostream>
using namespace std;

int addition(int num1, int num2);

main(){
    int number1, number2, result;
    cout << "Enter First Number: ";
    cin >> number1;
    cout << "Enter Second Number: ";
    cin >> number2;
    result = addition(number1, number2);
    cout << "Sum is: " << result;
}

int addition(int num1, int num2)
{
    int sum = num1 + num2;
    return sum;
}
```

Void Functions

We can also write a function that **doesn't take any input** and **doesn't return any output**.



```cpp
#include <iostream>
using namespace std;

void addition();

main(){
    addition();
}

void addition()
{
    int number1, number2;
    cout << "Enter First Number: ";
    cin >> number1;
    cout << "Enter Second Number: ";
    cin >> number2;
    int sum = number1 + number2;
    cout << "Sum is: " << sum;
}
```

Function Prototype

2 types of User-Defined Functions

We have studied 2 types of user-defined functions.

1. Value Returning Function
2. Void Function (which returns nothing)

```cpp
#include <iostream>
using namespace std;

int addition(int num1, int num2);

main(){
    int number1, number2, result;
    cout << "Enter First Number: ";
    cin >> number1;
    cout << "Enter Second Number: ";
    cin >> number2;
    result = addition(number1, number2);
    cout << "Sum is: " << result;
}

int addition(int num1, int num2)
{
    int sum = num1 + num2;
    return sum;
}
```

Value Returning Function

Which one is better?

```cpp
#include <iostream>
using namespace std;

void addition();

main(){
    addition();
}

void addition()
{
    int number1, number2;
    cout << "Enter First Number: ";
    cin >> number1;
    cout << "Enter Second Number: ";
    cin >> number2;
    int sum = number1 + number2;
    cout << "Sum is: " << sum;
}
```

Void Function

Property of Functions

The single-responsibility principle is a computer-programming principle that states that every function in a computer program should have responsibility over a single part of that program's functionality.

```cpp
#include <iostream>
using namespace std;

int addition(int num1, int num2);

main(){
    int number1, number2, result;
    cout << "Enter First Number: ";
    cin >> number1;
    cout << "Enter Second Number: ";
    cin >> number2;
    result = addition(number1, number2);
    cout << "Sum is: " << result;
}

int addition(int num1, int num2)
{
    int sum = num1 + num2;
    return sum;
}
```

✓ Value Returning Function

Which one is better?

```cpp
#include <iostream>
using namespace std;

void addition();

main(){
    addition();
}

void addition()
{
    int number1, number2;
    cout << "Enter First Number: ";
    cin >> number1;
    cout << "Enter Second Number: ";
    cin >> number2;
    int sum = number1 + number2;
    cout << "Sum is: " << sum;
}
```

Void Function

When the Void functions are used?

Previously, we have seen an example that value returning functions are better.

Then, the question is why and when Void functions are used?

When the Void functions are used?

Void functions can be used when we want to **print information** for the user to read.

For example,

```
void printName(string name)
{
    cout << "Username is: ", name;
}
```

When the Void functions are used?

Void functions can be used when we want to print information for the user to read.

For example,

```
void printMenu()
{
    cout << "*****Welcome*****";
    cout << "1. Login";
    cout << "2. Logout";
}
```

Learning Outcome

In this lecture, we learnt the difference between Void and Value Returning Functions.

Self Assessment

1. Write a program that asks a name and say hello. Use your own function, that receives a string of characters (name) and prints on screen the hello message.
   Hint: Make the void function

2. Make the function for the Header of your UAMS System.
   Hint: Make the function void.

Self Assessment

3. Create a function that takes three integer equal (a, b, c) and returns the amount of integers which are of equal value.
Your function must return 0, 2 or 3.

Test Cases:

- equal(3, 4, 3) → 2
- equal(1, 1, 1) → 3
- equal(3, 4, 1) → 0