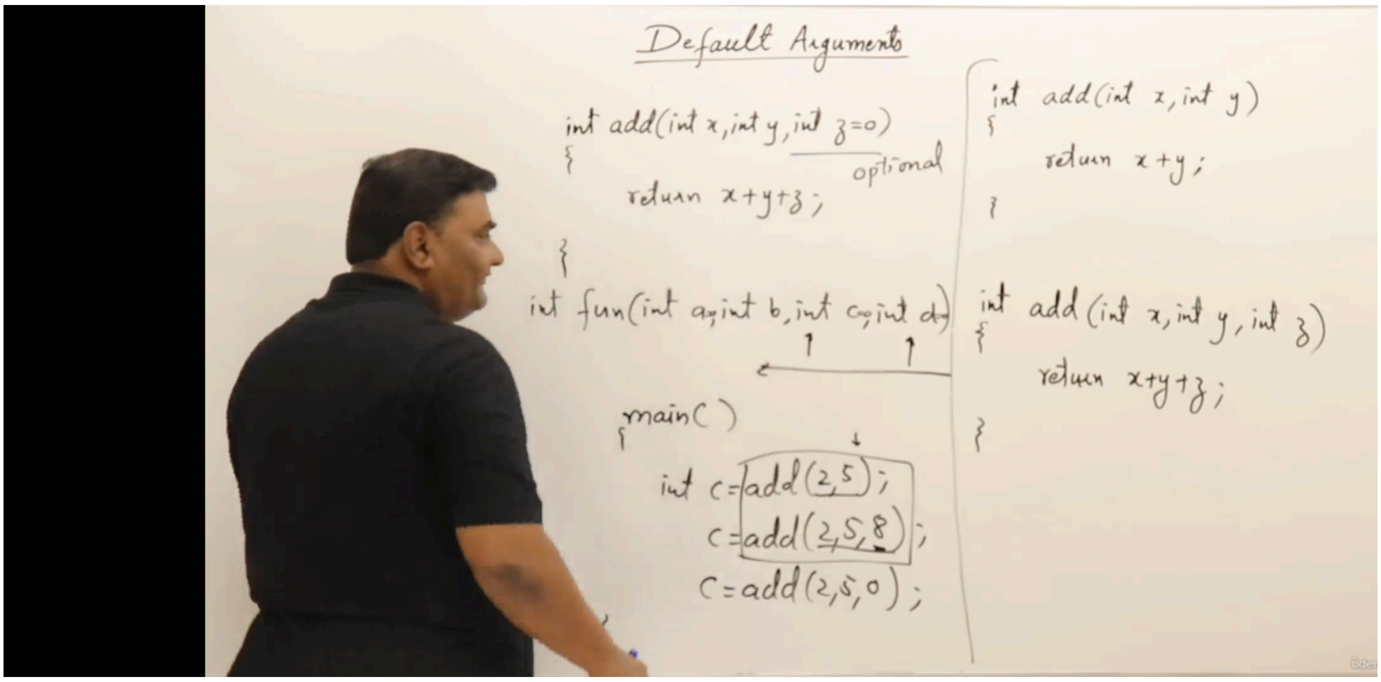

Default Arguments

```
int add(int x, int y, int z=0)
{                         optional
    return x+y+z;
}
int fun(int a, int b, int c, int d)

main()
{
    int c=add(2,5);
    c=add(2,5,8);
    c=add(2,5,0);
}
```

```
int add(int x, int y)
{
    return x+y;
}

int add(int x, int y, int z)
{
    return x+y+z;
}
```

Function Template - FAQ

- **max() function is giving error**

max() is a inbuilt function in C++.

Change the name to maxim()

- **Can we have a template function along with default argument?**

No.

- **class vs typename**

Both are same. You can use any one

- **Can we initialise template variable**

Yes. It should be initialised only with 0.

Default Arguments

- **Can a default argument function also be a template**

No.

- **Default values should be filled from which side**

Default values for formal arguments must be foibles from right side without skipping any parameter.

Function Overloading -FAQ

- **What is signature/prototype?**

The header of a function is called as signature or prototype.

Example:

int fun(int x,float y);

- **Two functions with same name. Are they overloaded ?**

Yes, they are overloaded functions if their parameters are different.

- **Is the return type considered in overloading?**

No.

- **Two functions with same name and parameters, but different return type. Are they overloaded?**

No. Return type is not considered in overloading.

Example:

These are not overloaded

int fun(int x, int y)

float fun(int x, int y)

- **Are these functions overloaded?**

int fun(int x, float y) and int fun(float x, int y)

Yes. They are overloaded

Functions - FAQ

- **Will the functions occupy space in memory?**

Yes, the machine code of a function is kept code section.

- **Will a function occupy space even if it is not called?**

Yes, if a function is defined in a program or included from library, it will occupy space in code section.

- **Where the memory for variable of a function is created?**

Memory for the variables used in a function is created in stack

- **When the memory for variables will be allocated?**

Memory for the variables will be allocated at runtime, when the function is called and deleted when function ends.

- **Is the memory for variables is allocated freshly for each call?**

For for each call of a function memory for the variables is created freshly in the stack.

- **What is return type of a function?**

When a function is called by passing parameters, it will compute and get the results. A function can return the result to a calling function.

Return type is the datatype of a value return by the function.

- **What is void?**

If a function is not returning any value then tis return type is mentioned as void.

●Difference between int main() and void main()

void main() means main function is not returning any value.

int main() means main function will return 0; 0 is a success code. The function have terminated successfully. main() will return the value to operating system, like windows.

In C++ int main() is standard.

Default Arguments

- Parameters of a function can have default values
- If a parameter is default then , passing its value is options
- Function with default argument can be called with variable number of argument
- Default values to parameters must be given from right side parameter
- Default arguments are much useful in constructors
- Default arguments are useful for defining overloaded functions

Example of Default Arguments

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

int sum(int a,int b,int c=0)
{
    return a+b+c;
}

int main()
{
    cout<<sum(10,20,3)<<endl;

    return 0;
}
```