Default argument to Function:

1. Only trailing(starting) argument can have default values. Only Middle or last argument can not have default value see:

2. the default parameter definition is usually specified in the function declaration:

**// f1(int,int=5,int);//Error**

**// f1(int=7,int=5,int);//Error**

**//void f2(int=5,int,int=0);//Error**

**Program:1**

**class A**

**{**

**public:**

**void f(int,int=5,int=0);we can call like: f(30)**

**// f1(int,int=5,int);//Error,can not call like: f(10, ,20);**

**//void f2(int=5,int,int=0);//Error can not call like: f( ,30,,20);**

**};**

**void A::f(int x,int y,int z)**

**{**

**cout<<x<<" "<<y<<" "<<z<<endl;**

**}**

**int main()**

**{**

**A a;**

**//a.f(10,20,30);//10 20 30**

**//a.f(10);//10 5 0**

**//a.f()//error**

**//a.f(10, ,30);//Error**

**return 0;**

**}**

**Note: specify value at declaration time or initialization time**

**1.**

int Add (int a, int b = 3);

int Add (int a, int b)

{

}

**2.**

int Add (int a, int b);

int Add (int a, int b = 3)

{

}

**Both work: Which better**

suppose:

**lib.h**

int Add(int a, int b);

**lib.cpp**

int Add(int a, int b = 3) {

...

}

**test.cpp**

#include "lib.h"

int main() {

Add(4);

}

The compilation of test.cpp will not see the default parameter declaration, and will fail with an error.

For this reason, the default parameter definition is usually specified in the function declaration:

**Default value to constructor:**

**We can speicify default value to constructor at declaration time or definition time also.**

**Program 1: at declaration time.**

**class A {**

**int y;**

**public:**

**A(int =30);//OR A(int x=3)**

**};**

**A::A(int x)**

**{**

**y=x;**

**cout<<"A consructor"<<y<<endl;**

**}**

**int main()**

**{**

**A a;**

**return 0;**

**}**

**Output:** **A consructor30**

**Program 2: initialization time**

**class A {**

**int y;**

**public:**

**A(int);**

**};**

**A::A(int x=30)**

**{**

**y=x;**

**cout<<"A consructor"<<y<<endl;**

**}**

**int main()**

**{**

**A a;**

**return 0;**

**}**