CSDN 学院

C++语言基础

迁者 - 贺利坚

http://blog.csdn.net/sxhelijian/

http://edu.csdn.net

本节主题：

函数模板

重载函数

●用重载函数，一个函数名，实现多种功能——清楚、简单。
●分别定义每一个函数——繁！烦！

```
#include <iostream>
using namespace std;
int max(int a,int b,int c);        //函数声明
double max(double a,double b,double c);
long max(long a,long b,long c);
int main( )
{
   int i1,i2,i3,i;
   cin>>i1>>i2>>i3;      //输入3个整数
   i=max(i1,i2,i3);       //求3个整数中的最大者
   cout<<"i_max="<<i<<endl;
   double d1,d2,d3,d;
   cin>>d1>>d2>>d3;    //输入3个双精度数
   d=max(d1,d2,d3);     //求3个双精度数中的最大者
   cout<<"d_max="<<d<<endl;
   long g1,g2,g3,g;
   cin>>g1>>g2>>g3;   //输入3个长整数
   g=max(g1,g2,g3);    //求3个长整数中的最大者
   cout<<"g_max="<<g<<endl;
}
```

```
int max(int a,int b,int c)
{
   if(b>a) a=b;
   if(c>a) a=c;
   return a;
}
double max(double a,double b,double c)
{
   if(b>a) a=b;
   if(c>a) a=c;
   return a;
}
long max(long a,long b,long c)
{
   if(b>a) a=b;
   if(c>a) a=c;
   return a;
}
```

函数模板(function template)：替代同体重载函数的多次定义

- 所谓函数模板，实际上是建立一个**通用函数**，其函数类型和形参类型可以不具体指定，而是用一个**虚拟的类型**来代表。
- 凡是函数体相同的函数都可以用这个模板来代替，不必定义多个函数，只需在模板中定义一次即可。
- 在调用函数时系统会根据实参的类型来取代模板中的虚拟类型，从而实现了不同函数的功能。

使用用函数模板

```
#include <iostream>
using namespace std;
template<typename T>   //模板声明，其中T为虚拟类型
T max(T a,T b,T c)   //定义一个通用函数，用T作虚拟的类型名
{
   if(b>a) a=b;
   if(c>a) a=c;
   return a;
}
```

```
int main( )
{
   int i1=185,i2=-76,i3=567;
   double d1=56.87,d2=90.23,d3=-3214.78;
   long g1=67854,g2=-912456,g3=673456;
   cout<<"i_max="<<max(i1,i2,i3)<<endl;
   cout<<"f_max="<<max(d1,d2,d3)<<endl;
   cout<<"g_max="<<max(g1,g2,g3)<<endl;
   cout<<"c_max="<<max('1','a','A')<<endl;
   return 0;
}
```

调用时，分别用相应的数据类型代替模板函数中的T

定义函数模板

- 先进行模板声明，再定义函数

  template <typename T>

  template <class T>

- 虚拟类型名即可以被用于函数定义
- 在对程序进行编译时，编译系统会将函数名与模板相匹配，将实参的类型取代了函数模板中的虚拟类型T。
- 类型参数可以不只一个，可以根据需要确定个数。如

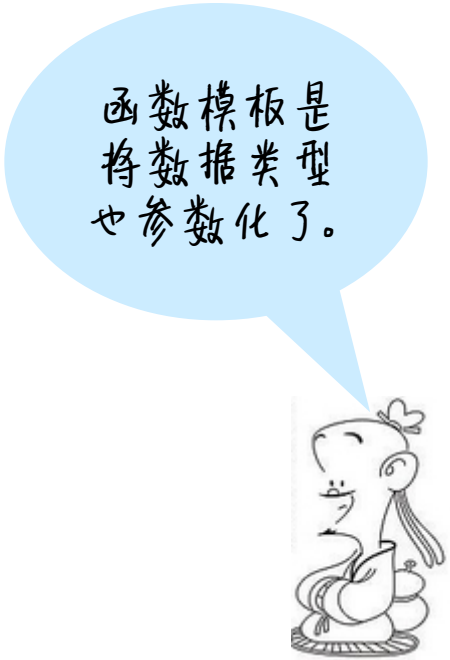
  template <class T1, typename T2>

```
template<typename T>
T max(T a,T b,T c)
{
  if(b>a) a=b;
  if(c>a) a=c;
  return a;
}

  cout<<max(i1,i2,i3);
  cout<<max(d1,d2,d3);
  cout<<max(g1,g2,g3);
```

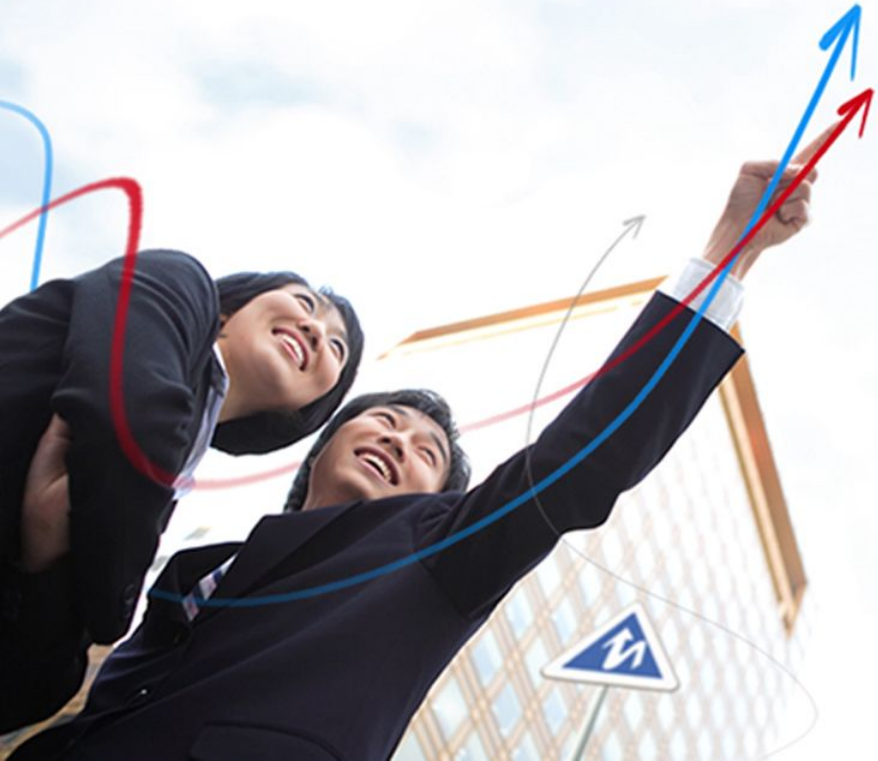
体会：参数带来的灵活性

```
#include <stdio.h>
void printchs(int m, char ch)
{
   int j;
   for (j=1; j<=m; ++j)
      putchar(ch);
}
int main()
{
   int i;
   for(i=1; i<=6; ++i)
   {
      printchs(6-i,' ');
      printchs(2*i-1,'*') ;
      printf("\n");
   }
   return 0;
}
```

```
template<typename T>
T max(T a,T b,T c)
{
  if(b>a) a=b;
  if(c>a) a=c;
  return a;
}

 cout<<max(i1,i2,i3);
 cout<<max(d1,d2,d3);
 cout<<max(g1,g2,g3);
```



CSDN 学院



THANKS

本课程由 迁者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：http://ems.csdn.net/
人才服务：http://job.csdn.net/
CTO俱乐部：http://cto.csdn.net/
高校俱乐部：http://student.csdn.net/
程序员杂志：http://programmer.csdn.net/

CODE平台：https://code.csdn.net/
项目外包：http://www.csto.com/
CSDN博客：http://blog.csdn.net/
CSDN论坛：http://bbs.csdn.net/
CSDN下载：http://download.csdn.net/