



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY

C++程序设计

Programming in C++



1011018

主讲：魏英，计算机学院

函数的设计

- ◆ 3、函数模板
- ◆ 4、库函数的调用

- ▶ 函数模板（function template）是一个独立于类型的函数，可作为一种模式，产生函数的特定类型版本。

```
int abs( int x )
```

```
{ return x<0?-x:x; }
```

```
double abs( double x )
```

```
{ return x<0?-x:x; }
```

- ▶ 使用函数模板可以设计通用型的函数，这些函数与类型无关并且只在需要时自动实例化，从而形成“批量型”的编程方式。

- ▶ 函数模板定义的语法形式为：

```
template<模板形参表>返回类型 函数名(形参列表)
{
    函数体
}
```

```
template<typename T>
T abs(T x)
{ return x<0?-x:x; }
```

```
int main()
{ int n=-5; double d=-5.5;
  cout<<abs(n)<<','<<abs(d)<<endl;
  return 0;
}
```

- ▶ 模板形参表（template parameter list）是用一对尖括号括< >括起来的一个或多个模板形参的列表，不允许为空，形参之间以逗号分隔，其形式有两种。
- ▶ ①第一种形式

`typename` 类型参数名1, `typename` 类型参数名2,

- ▶ ②第二种形式

`class` 类型参数名1, `class` 类型参数名2,

10.3 函数模板

【例10.4】函数模板举例。

```
1  #include <iostream>
2  using namespace std;
3  template <class T> T add(T a, T b)
4  {
5      return a+b;
6  }
7  int main()
8  {
9      cout<<"int_add="<<add(10,20)<<endl; //生成整型版本的add函数
10     cout<<"double_add="<<add(10.2,20.5)<<endl; //生成实型版本的add函数
11     cout<<"char_add="<<add('A','\2')<<endl; //生成字符型版本的add函数
12     cout<<"int_add="<<add(100,200)<<endl;
13     return 0;
14 }
```

- ▶ 所谓**库函数**是指事先由程序员编制好的函数。
- ▶ 多数情况下，基于各种理由，如保护知识产权，这些库函数仅提供二进制形式的目标代码给调用者链接，却没有提供源码形式的函数定义。

- ▶ (1) 在程序中添加库函数声明
- ▶ 多数库函数将自己的函数原型和特殊数据等放在头文件（.h）中，所以应首先使用文件包含命令将这些头文件包含到程序中。例如欲使用数学库函数，文件包含命令为：
 - ▶ `#include <cmath>`
- ▶ 从而使得程序有函数声明，例如：
 - ▶ `y=sin(x);`
 - ▶ 调用就能够通过编译。

- ▶ (2) 将库函数目标代码连接到程序中。
- ▶ 在连接时，例如使用了sin函数，就必须要有sin函数的实现代码才能生成可执行文件，否则连接出错。要将库函数的目标代码能够连接到程序中，主要是配置好开发环境的相关参数，然后由连接器处理。
- ▶ 标准库函数的连接在开发环境中是默认的，一般可以不用特别设置。

CP 程序设计