



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

作用域、生命期和程序的组织结构

- ◆ 4、内部函数与外部函数.....
- ◆ 5、头文件与工程文件.....
- ◆ 6、多文件编译.....

- ▶ 内部函数
- ▶ 函数本质上是全局的，在多文件的程序中，在连接时会检查函数在全局作用域内是否名字唯一，如果不是则出现连接错误。
- ▶ 在函数定义前加上static修饰，则函数称为内部函数，定义形式为：

```
static 返回类型  函数名(形式参数列表)
{
    函数体
}
```

- ▶ 按照前面的实体可见规则，内部函数仅在包含它的文件中有效。
- ▶ 之所以使用内部函数的原因是该函数逻辑上仅限定在一个文件中使用，其他文件不会用到。而且希望连接检查时永远不可能出现该函数名不唯一的连接错误，这在多人编写同一个程序的软件开发模式中是常用的策略。

- ▶ 外部函数
- ▶ 在函数定义前加上extern声明，则函数称为外部函数，定义形式为：

```
extern 返回类型  函数名(形式参数列表)
{
    函数体
}
```

- ▶ C++中所有的函数本质上都是外部函数。因此，上面的extern都可以省略。

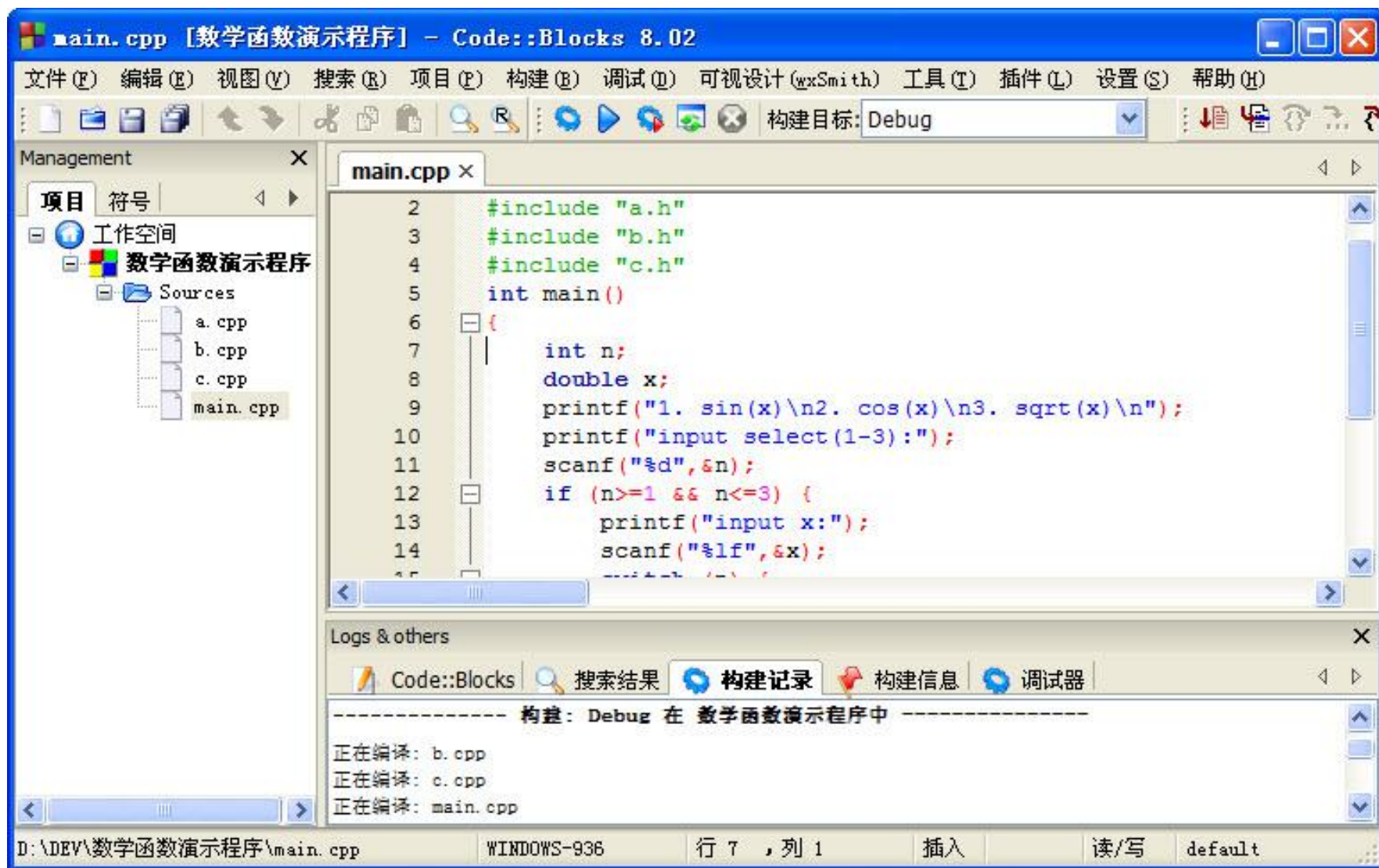
- ▶ 为什么要使用头文件呢？
- ▶ 我们现在已经知道，如果是多文件结构程序，欲在文件中调用别的文件中的函数，需要有函数的声明，而且每个文件均是如此。如果是函数声明比较多的情况下，在每个文件都写上函数声明不是好办法，很难管理。例如：某个函数定义有变动，那么所有含有这个函数声明的调用文件都需要找出来，逐一修改。

- ▶ 使用头文件可以解决这个问题，其工作原理是通过将每个源文件中外部函数的函数声明等信息集中写到一个文件中，称为头文件（有别于源文件），而别的源文件只需用文件包含命令将这个头文件包含，则编译时编译器自然就有了函数声明。

- ▶ 工程文件
- ▶ 多文件结构程序在编译时需要工程文件来管理，不同的编译器有不同的工程文件格式。

12.5 头文件与工程文件

图12.1 CodeBlocks工程文件管理



12.6 多文件编译

【例12.4】编写sin(x)、cos(x)、sqrt(x)数学函数演示程序。

```
1  //a.h文件
2  double fsin(double x);
```

```
1  //a.cpp文件
2  #include <cmath>
3  #include "main.h"
4  double fsin(double x)
5  {
6      return sin(x*PI/180.0);
7  }
```

12.6 多文件编译

```
1  //⑥b.h文件
2  double fcos(double x);
```

```
1  //⑤b.cpp文件
2  #include <cmath>
3  #include "main.h"
4  double fcos(double x)
5  {
6      return cos(x*PI/180.0);
7  }
```

12.6 多文件编译

```
1  //⑧c.h文件
2  double fsqrt(double x);
```

```
1  //⑦c.cpp文件
2  #include <cmath>
3  double fsqrt(double x)
4  {
5      return sqrt(x);
6  }
```

12.6 多文件编译

```
1 //②main.h文件
2 #define PI 3.1415926
```

```
1 //①main.cpp文件
2 #include <iostream>
3 using namespace std;
4 #include "a.h"
5 #include "b.h"
6 #include "c.h"
7 int main()
8 {
9     int n; double x;
10     cout<<"1. sin(x)\n2. cos(x)\n3. sqrt(x)\ninput select(1-3):";
11     cin>>n; //选择
12     cout<<"input x: "; cin>>x;
```

12.6 多文件编译

```
13      switch (n) {  
        //根据n分别调用a.c、b.c、c.c的函数  
14      case 1: cout<<"sin="<<fsin(x)<<endl; break;  
15      case 2: cout<<"cos="<<fcos(x)<<endl; break;  
16      case 3: cout<<"sqrt="<<fsqrt(x)<<endl; break;  
17      }  
18      return 0;  
19  }
```

运行结果:

```
3 ✓  
input x: 9 ✓  
sqrt=3
```

CP 程序设计