

程序调试方法简介

江苏大学计算机学院计算机科学系

主讲教师：王新宇

基本手段

● 断点

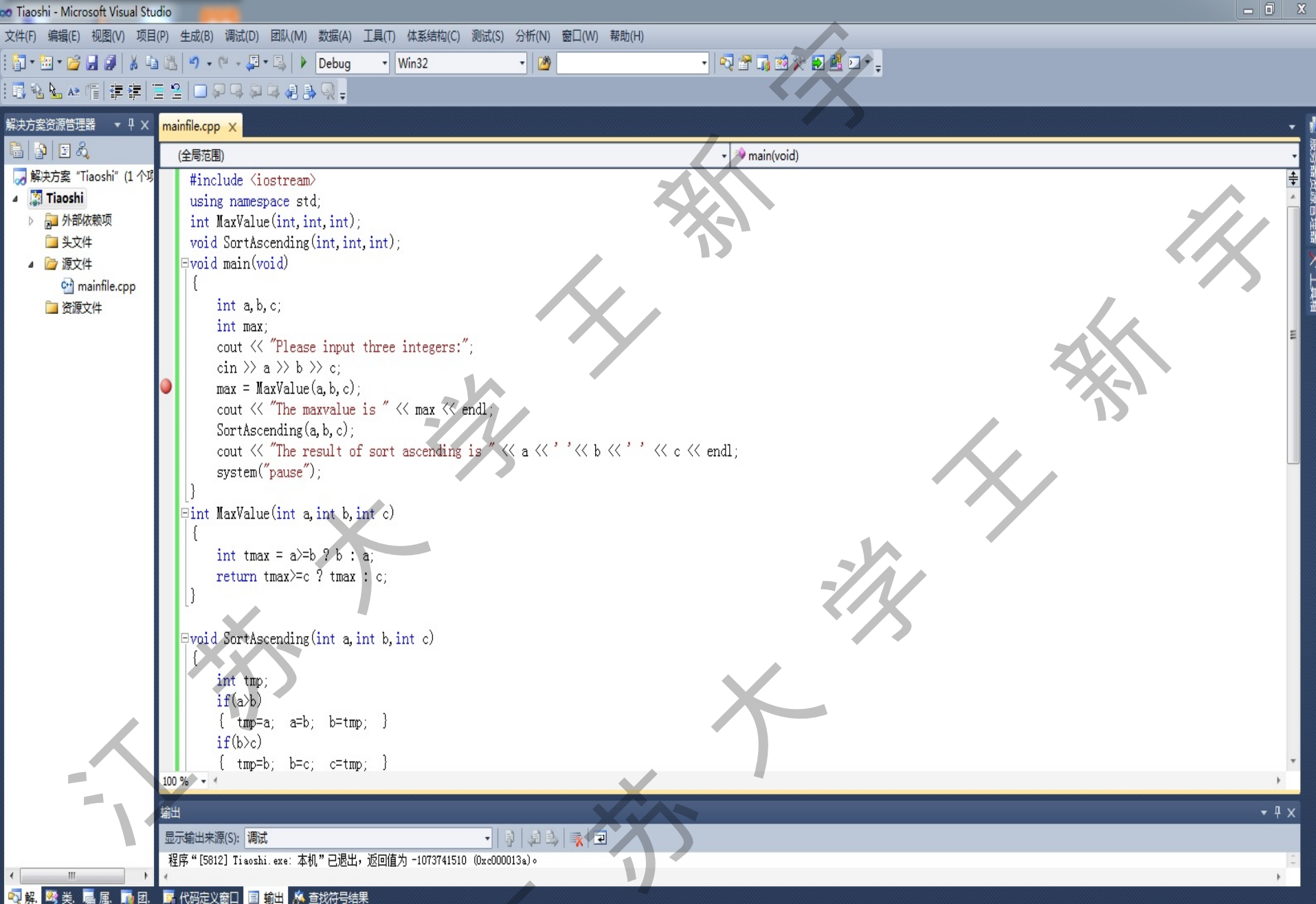
(1) 设置断点

将输入光标位置移到要设置断点的行，按下**F9**，
即在该行设置了断点。

(2) 取消断点

将输入光标位置移到要取消断点的行，按下**F9**，
即将该行设置的断点取消。

用途：设置断点后，按下**F5**执行程序，程序将在断点处暂停，程序员可以进行干预。



按下F9在光标所在行设置断点

Tiaoshi (正在调试) - Microsoft Visual Studio

文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 团队(M) 数据(A) 工具(T) 体系结构(C) 测试(S) 分析(N) 窗口(W) 帮助(H)

Debug Win32

进程: [3872] Tiaoshi.exe 线程: [928] 主线程 堆栈帧: Tiaoshi.exe!main() 行 11

类视图

mainfile.cpp

(全局范围) main(void)

```
#include <iostream>
using namespace std;
int MaxValue(int,int,int);
void SortAscending(int,int,int);
void main(void)
{
    int a,b,c;
    int max;
    cout << "Please input three integers:";
    cin >> a >> b >> c;
    max = MaxValue(a,b,c);
    cout << "The maxvalue is " << max << endl;
    SortAscending(a,b,c);
    cout << "The result of sort ascending is " << a << ' ' << b << ' ' << c << endl;
    system("pause");
}
int MaxValue(int a,int b,int c)
{
    int tmax = a>b ? b : a;
    return tmax>=c ? tmax : c;
}
void SortAscending(int a,int b,int c)
{
    int tmp;
    if(a>b)
```

红色圆表示该行有断点，黄色箭头表示程序执行到的位置（该语句即将被执行）

监视 1

| 名称 | 值 | 类型 |
|----|---|----|
| | | |

调用堆栈

| 名称 | 语言 |
|--|-----|
| Tiaoshi.exe!main() 行 11 | C++ |
| Tiaoshi.exe!_tmainCRTStartup() 行 555 + 0 | C |
| Tiaoshi.exe!mainCRTStartup() 行 371 | C |

解决方案资源管理器 类视图 自动窗口 局部变量 线程 模块 监视 1 调用堆栈 断点 输出

按下F5执行程序，程序将在断点处暂停



● 单步执行

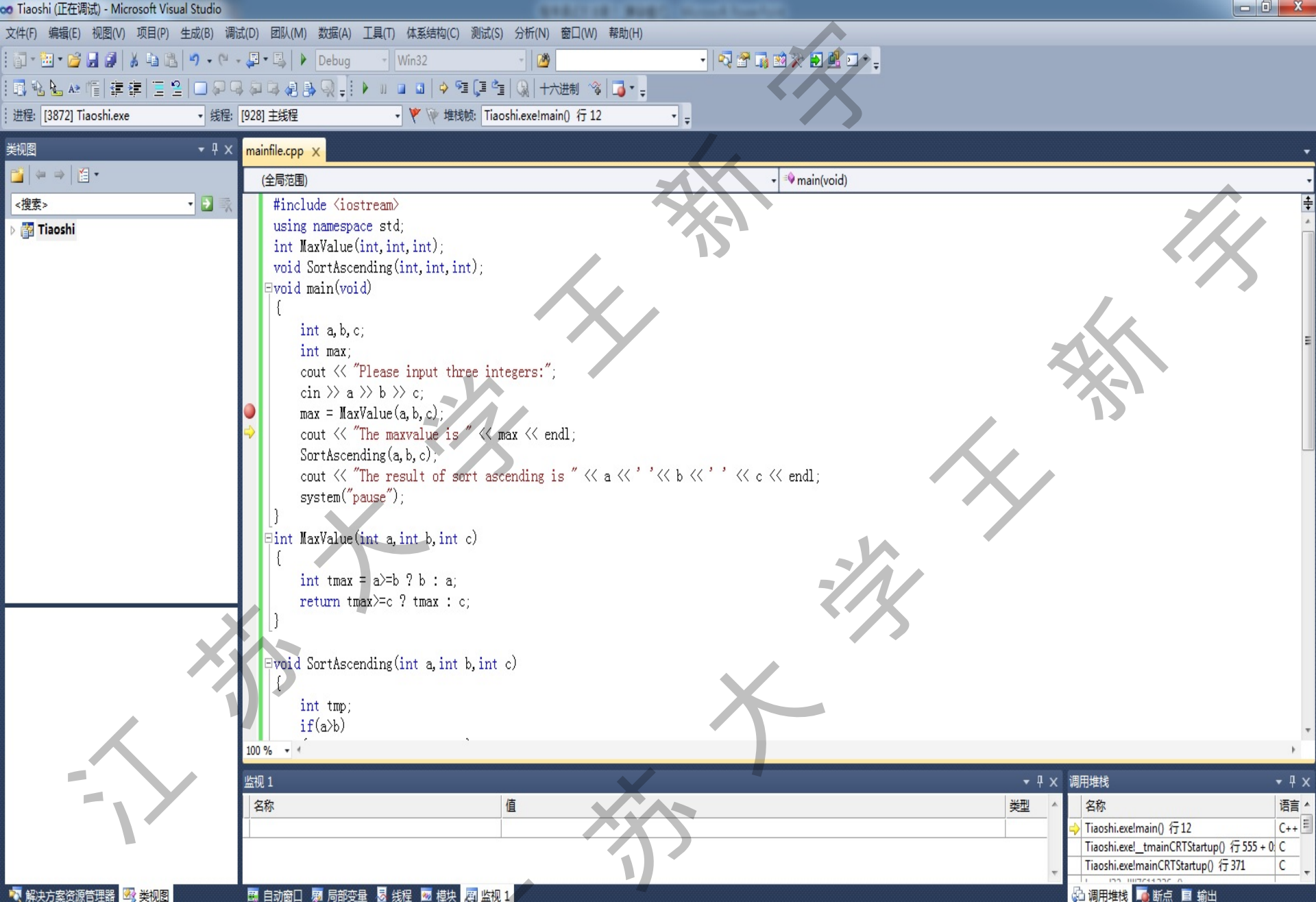
按下**F10**或**F11**可以单步执行程序，它们的区别在于对函数调用语句的处理不同。

F10：将函数调用作为一条普通语句执行，指示程序执行过程的箭头不会进入被调函数中。

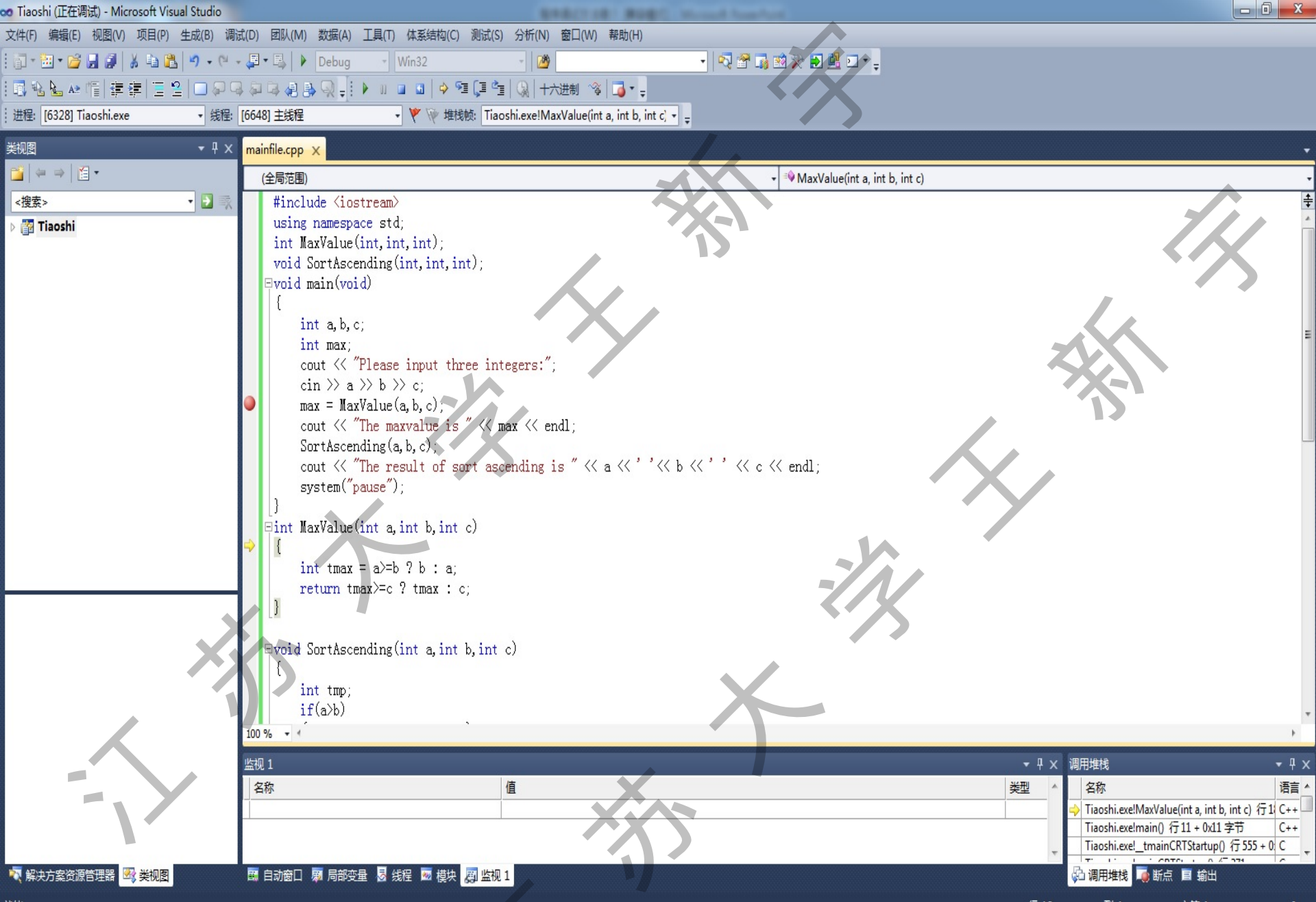
F11：指示程序执行过程的箭头会进入被调函数中。

用途：通过单步执行，可以跟踪程序每一步的执行结果。






F10: 将函数调用作为一条普通语句执行



F11: 指示程序执行过程的箭头进入被调函数



● 观察程序运行中变量的值

当程序执行遇到断点暂停或在单步执行时，在监视窗口设置要观察的变量，即可观察这些变量在程序执行过程中的值的变化情况。

用途：观察变量在程序执行时的实际值是否正确，从而确定错误位置。



Tiaoshi (正在调试) - Microsoft Visual Studio

文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 团队(M) 数据(A) 工具(T) 体系结构(C) 测试(S) 分析(N) 窗口(W) 帮助(H)

Debug Win32

进程: [6328] Tiaoshi.exe 线程: [6648] 主线程 堆栈帧: Tiaoshi.exe!MaxValue(int a, int b, int c)

类视图

mainfile.cpp

(全局范围) MaxValue(int a, int b, int c)

```
#include <iostream>
using namespace std;
int MaxValue(int,int,int);
void SortAscending(int,int,int);
void main(void)
{
    int a,b,c;
    int max;
    cout << "Please input three integers:";
    cin >> a >> b >> c;
    max = MaxValue(a,b,c);
    cout << "The maxvalue is " << max << endl;
    SortAscending(a,b,c);
    cout << "The result of sort ascending is " << a << ' ' << b << ' ' << c << endl;
    system("pause");
}
int MaxValue(int a,int b,int c)
{
    int tmax = a>b ? b : a;
    return tmax>c ? tmax : c;
}
void SortAscending(int a,int b,int c)
```

监视窗口, 第一栏: 变量, 第二栏: 变量当前的实际值

监视 1


| 名称 | 值 | 类型 |
|----|----|-----|
| a | 11 | int |
| b | 5 | int |
| c | 8 | int |

调用堆栈

| 名称 | 语言 |
|--|-----|
| Tiaoshi.exe!MaxValue(int a, int b, int c) 行 11 | C++ |
| Tiaoshi.exe!main() 行 11 + 0x11 字节 | C++ |
| Tiaoshi.exe!_tmainCRTStartup() 行 555 + 0 | C |
| Tiaoshi.exe!mainCRTStartup() 行 371 | C |
| kernel32.dll!7611336a() | |
| [下面的框架可能不正确和/或缺失, 没有为...] | |

解决方案资源管理器 类视图 自动窗口 局部变量 线程 模块 监视 1 调用堆栈 断点 输出

监视窗口观察变量当前值




程序调试基本步骤

1. 估计代码中发生逻辑错误的行，在该行设置断点。
 2. 运行程序到该断点暂停。
 3. 观察程序运行到断点时程序中变量的取值是否正确，如果正确，则表明断点之前的程序是没有问题的，则继续第4步；否则，表明断点之前的程序有错误，则返回第1步。
 4. 单步执行，观察单步执行后程序变量的取值，如果正确表明该断点所在行没有问题，取消该断点，返回第1步；如果错误，表明该断点所在行有错，则找到错误所在行，仔细检查和分析错误发生的原因；如果该行代码是对函数的调用，则需要在该函数中设置断点，返回第1步。
 5. 解决该错误后，重新运行程序，如果仍然有错，则返回第1步继续调试。
-

参考程序

```
#include <iostream>
using namespace std;
int MaxValue(int,int,int);
void SortAscending(int,int,int);
void main(void)
{
    int a,b,c;
    int max;
    cout << "Please input three integers:";
    cin >> a >> b >> c;
    max = MaxValue(a,b,c);
    cout << "The maxvalue is " << max << endl;
    SortAscending(a,b,c);
    cout << "The result of sort ascending is " << a << ' ' << b << ' ' << c <<
    endl;
    system("pause");
}
```




```
int MaxValue(int a,int b,int c)
```

```
{  
    int tmax = a>=b ? b : a;  
    return tmax>=c ? tmax : c;  
}
```

```
void SortAscending(int a,int b,int c)
```

```
{  
    int tmp;  
    if(a>b)  
    { tmp=a; a=b; b=tmp; }  
    if(b>c)  
    { tmp=b; b=c; c=tmp; }  
    if(a>b)  
    { tmp=a; a=b; b=tmp; }  
}
```



实践才是硬道理！