

# ***C++ Programming***

---

## ***Experiment 2***

Wei Liu (刘威)

Dept. of Electronics and Information Eng.

Huazhong University of Science and Technology

March 2015



# 实验二

---

- ➡ ● 学习程序的录入与调试
- 完成上机实验题



# 程序的录入

- 程序录入： **先框架后细节，代码缩进！**
  - 先录入程序的主题框架部分，比如空的main函数
  - 先输入()**{ }**等成对的控制符号，然后在编制内部的代码
  - 已录入的代码编译无误后，**再编制执行特定功能的代码**
- 用 **Tab**键缩进代码，多用空格区分代码
- 保持良好的代码缩进风格，以便于代码查看



# 程序的调试

- 程序编译出错的解决方案
  - 编译未能通过，说明开发环境认为C语法检查没有通过
  - 1. 观察编译的错误信息，一般这些错误信息会对应到相应的错误行上；
  - 2. 如果相应行没有问题，需要进行前后语句的排查，可以用 `/* */` 块注释分区域编译，来逐步确定错误语句的区域



# 程序的调试

- 程序编译通过，但是代码运行错误的解决方案
  - 编译能通过，说明C语法检查通过；而程序运行错误，则是程序逻辑出了问题
  - 需要用开发环境的调试debug功能跟踪相应代码的执行过程，观察程序的执行流程、变量的值是否符合预期设计目标



# 实验二

---

- 学习程序的录入与调试
  - 1. 跟踪代码的执行
  - 2. 调试一个错误的程序
- 通过上机完成课后练习题
  - 3. 教材题目
- 完成上机实验题
  - 4. 编写一个计算星期几的程序



# 1. 跟踪代码的执行

- 建立一个工程，输入fig02\_05.c 的代码，跟踪该代码的执行

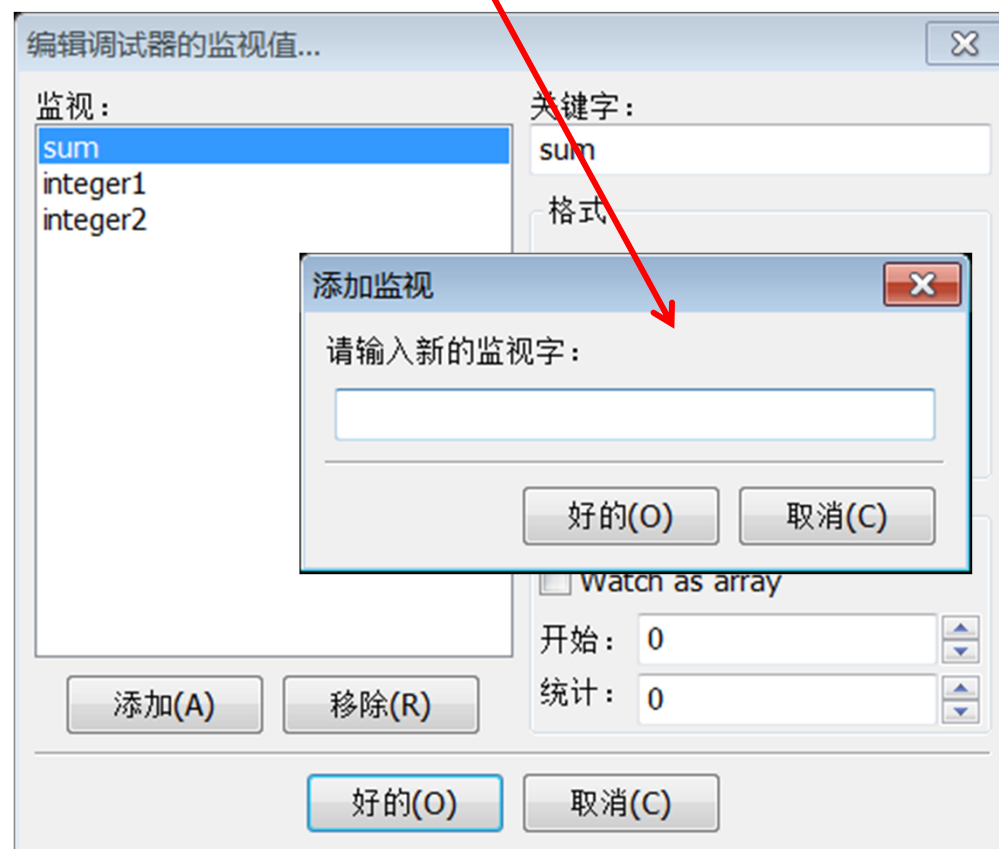
```
1  #include <stdio.h>
2
3  int main( void )
4  {
5      int integer1, integer2, sum ;
6
7      printf("Enter first integer\n");
8      scanf("%d", &integer1);
9
10     printf("Enter second integer\n");
11     scanf("%d", &integer2);
12
13     sum = integer1 + integer2;
14
15     printf("Sum is %d\n", sum);
16
17     return 0;
18 }
```

# 添加对变量运行值的监视

Build(B) 调试(D) wxSmith 工具(T) 插件(L) 设置(S) 帮助(H)



输入变量名称







# 添加程序运行时的断点

断点设置成功  
的行标记

```
1  #include <stdio.h>
2
3  int main( void )
4  {
5      int integer1, integer2, sum ;
6
7      printf("Enter integer1: ");
8      scanf("%d", &integer1);
9
10     printf("Enter integer2: ");
11     scanf("%d", &integer2);
12
13     sum = integer1 + integer2;
14     printf("Sum: %d\n", sum);
15     return 0;
16 }
17
18
19
```

- 切换断点
- 执行到光标处
- 插入
- 切换到头/源文件
- 编辑
- 书签
- 折叠
- Format this file (AStyle)
- Add Todo item...
- Split view
- Configure editor...
- Properties...
- 从项目里移除文件

在需要设置断点的行点击鼠标右键，“切换断点”



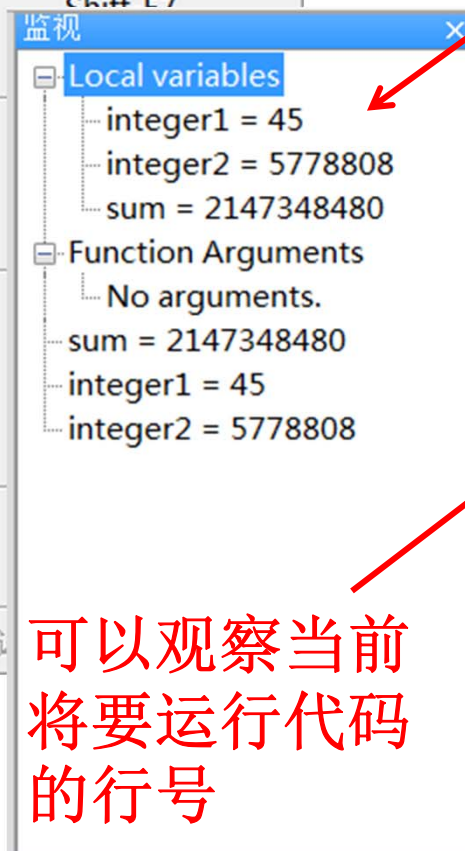
# 跟踪程序的运行

B) 调试(D) wxSmith 工具(T) 插件(L) 设置(S) 帮助(H)

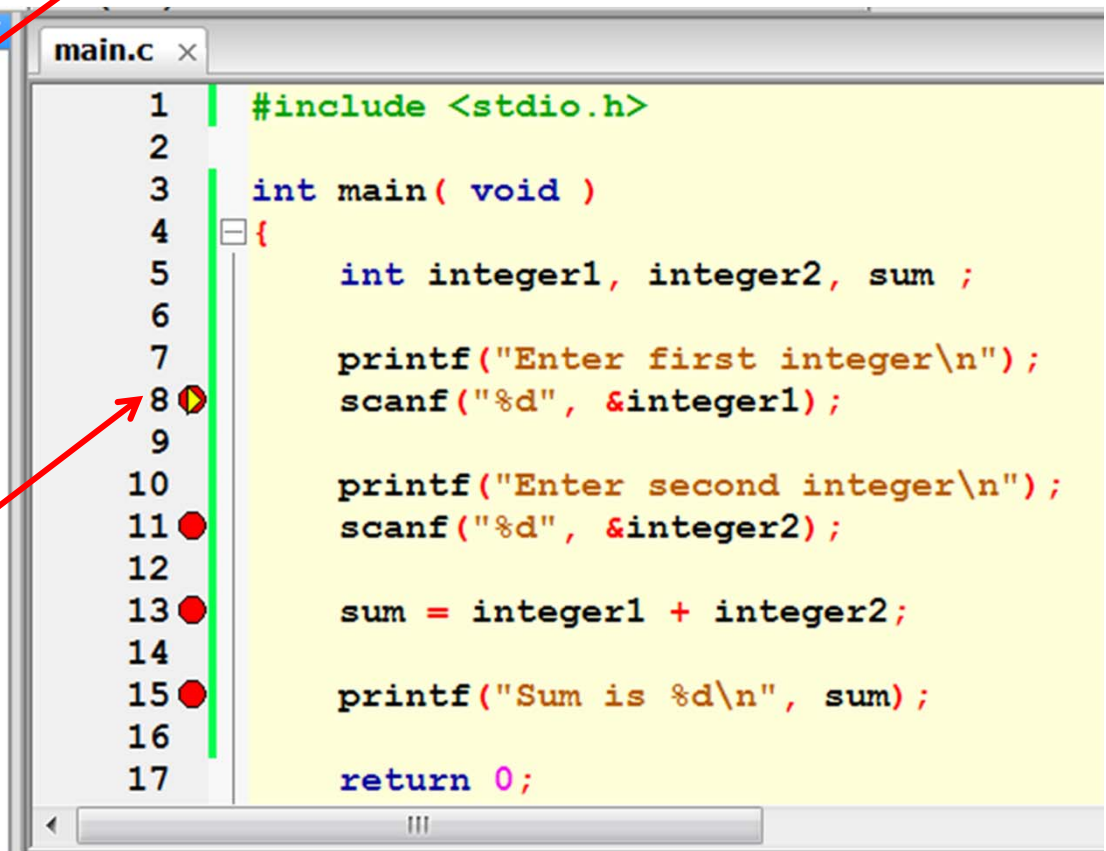


点击调试的“开始”

出现“监视”窗口，可以观察变量的值



可以观察当前  
将要运行代码  
的行号





# 跟踪程序的运行



点击调试的“执行到下一行”

在命令行窗口，  
输入12

```
E:\CodeCpp\test\sumtwo\bin\Debug\sumtwo.exe
Enter first integer
12
```

监视

- Local variables
  - integer1 = 12
  - integer2 = 6303096
  - sum = 2147348480
- Function Arguments
  - sum = 2147348480
  - integer1 = 12
  - integer2 = 6303096

main.c

```
1  #include <stdio.h>
2
3  int main( void )
4  {
5      int integer1, integer2, sum ;
6
7      printf("Enter first integer\n");
8      scanf("%d", &integer1);
9
10     printf("Enter second integer\n");
11     scanf("%d", &integer2);
12
13     sum = integer1 + integer2;
14 }
```

监视窗口可以看到integer1的值

程序执行到下一个断点

## 调试工具栏说明



## 开始调试



## 调试执行到下一个断点处



## 调试执行到下一行，不进入子函数



## 调试执行到下一条语句



## 调试进入子函数



## 调试离开子函数



## 结束调试



## 思考题

- 变量integer1、integer2、sum在定义之后的初始值是什么？
- 变量integer1、integer2、sum的值分别是什么时候改变的？
- 如果没有改变过变量integer1、integer2、sum的值，直接输出将会得到什么结果？





## 2. 调试一个错误的程序

- 建立一个工程，输入以下程序，通过跟踪调试判断该代码错误在哪里？

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a, b;
7      a = 1;
8      b = 2;
9
10     if ( a = b)
11     {
12         printf("Hello world!\n");
13     }
14
15     return 0;
16 }
```

E:\CodeCpp\test\iferror\bin\Debug\iferror.exe

Hello world!



### 3. 教材试题

---

- 2.7 找错误，把错误现象记录下来
- 2.19 计算两个数的各种运算结果
  - 使用 if 语句
- 2.20 计算圆的特征
  - 使用浮点数



## 4. 编写一个计算星期几的程序

- 运用所学到的知识，编写一个名为weekday.c的程序，计算2015年1月到3月的某一天是星期几
  - 提示：假设用户输入的月、日信息都是合理的，每个星期从周一开始，已知2015年1月1日是星期四，可以用取模运算

```
Please input month (1-12):  
3  
Please input day (1-31):  
5  
  
2015-3-5 is Thursday.
```





## 4. 编写一个计算星期几的程序

- 扩展问题1
  - 把程序支持的计算范围扩展到1月-12月
- 扩展问题2
  - 在输出时明确输出星期几

```
Please input month (1-12):  
3  
Please input day (1-31):  
5  
  
2015-3-5 is No.4 day in that week.  
Thursday.
```

```
Please input month (1-12):  
12  
Please input day (1-31):  
31  
  
2015-12-31 is No.4 day in that week.  
Thursday.
```

# 谢谢!

---



刘威 副教授

互联网技术与工程研究中心  
华中科技大学电子与信息工程系

Email: [liuwei@hust.edu.cn](mailto:liuwei@hust.edu.cn)