燕山大學
YANSHAN UNIVERSITY

C++面向对象程序设计

实验指导书

（1）编程环境的熟悉及简单 C++程序的编制

燕山大学软件工程系

目　录

实验 1 编程环境的熟悉及简单 C++程序的编制

1.1 时间安排

本实验安排 2 个实验课时。

1.2 实验目的和要求

1．熟悉 DEV C++编程环境，编制简单 C++程序并运行，熟悉 C++的编辑、 编译、连接、运行、断点调试等过程。

2．掌握 C++数据类型，熟悉如何定义和使用常量和变量，以及对它们赋值的方法。

3．学会使用 C++的有关算术运算符及表达式，特别是自加（++）和自减（--）运算符的使用。

4．分支和循环结构的使用

1.3 实验报告的撰写要求

将实验任务中红色字体的题目的构思过程、源码、运行结果（截图）、心得体会等内容按要求填写，详见实验报告模板。

1.4 实验内容

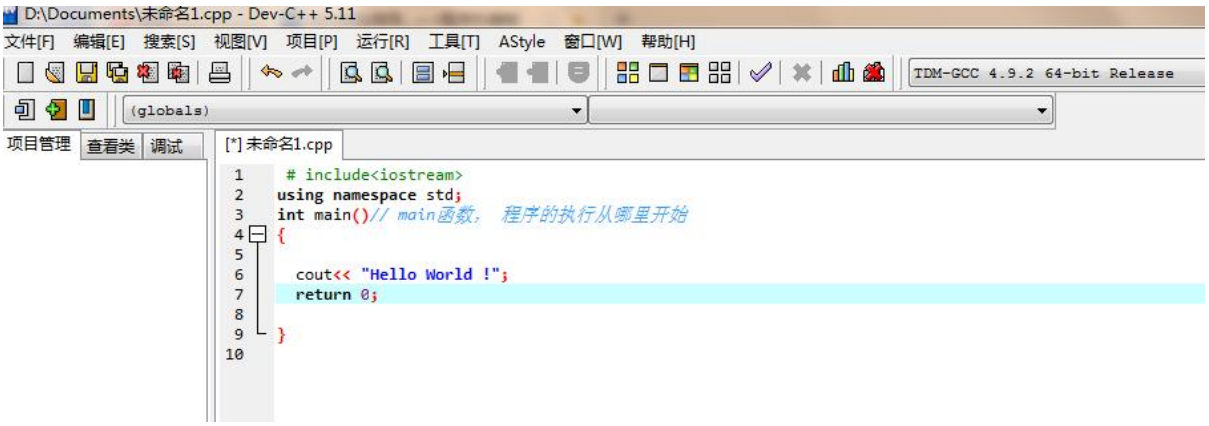
1.4.1 熟悉 DEV C++的编程环境

下面让我们用 DEV C++来做一个控制台的 HelloWorld 程序吧。DEV C++里面既可以单独编译一个.cpp 或者一个.c 文件，也可以创建一个项目。

1.4.1.1 直接新建一个.cpp 文件

1、点击工具栏“文件”—“新建”—“源代码”

2、我们编写一段代码在控制台输出一句话："Hello World !"。



这里需要注意：

（1）上述显示“Hello World ！”的简单 C++程序中“// main 函数， 程序的执行从哪里开始 ”是注释。注释用于显示有关程序的其他信息。注释不包含任何编程逻辑。当编译器遇到注释时，编译器将跳过该行代码。在 C++中，任何以'//'开头且不带引号或/*...*/之间的行都是注释。

（2）#include ：在 C++中，所有以#号（#）符号开头的行都称为指令，并由预处理器（由编译器调用的程序）处理。所述的#include 指令告诉编译器包括文件和#include<iostream> 。它告诉编译器包括标准库 iostream 文件，该文件包含所有标准输入/输出库函数的声明。

（3）using namespace std ：这用于将 std 命名空间的整体导入到程序的当前命名空间中。

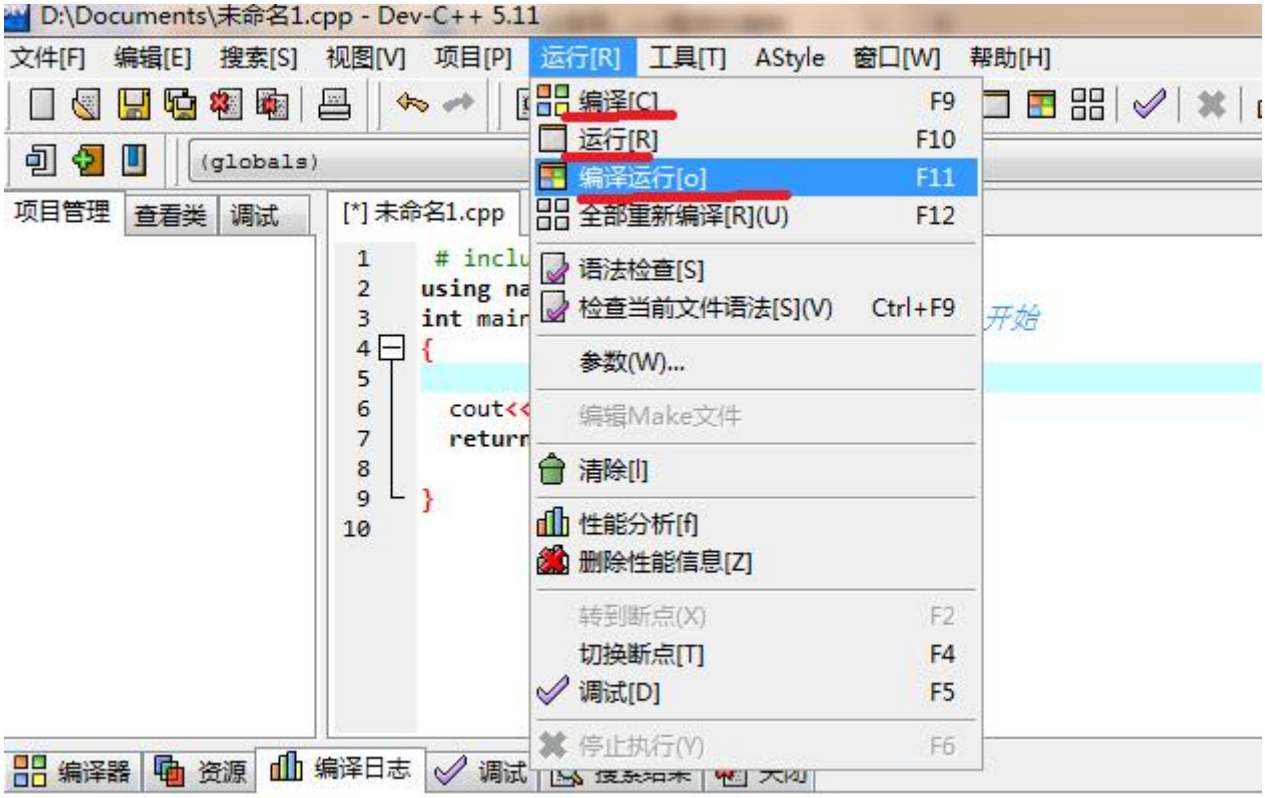
（4）int main ：此行用于声明一个名为“ main”的函数，该函数返回整数类型的数据。函数是一组旨在执行特定任务的语句。每个 C ++程序的执行都从 main 函数开始，无论该函数位于程序中的哪个位置。因此，每个 C++程序都必须具有 main 函数。

（5）cout<<“Hello World ！”; ：此行告诉编译器在屏幕上显示消息“ Hello World ！”。该行在 C++中称为语句。每个语句都旨在执行某些任务。分号“;” 用于结束语句。语句末尾的分号字符用于指示语句在此处结束。std :: cout 用于标识标准字符输出设备，通常是桌面屏幕。后面跟着字符“ <<”的所有内容都会显示在输出设备上。

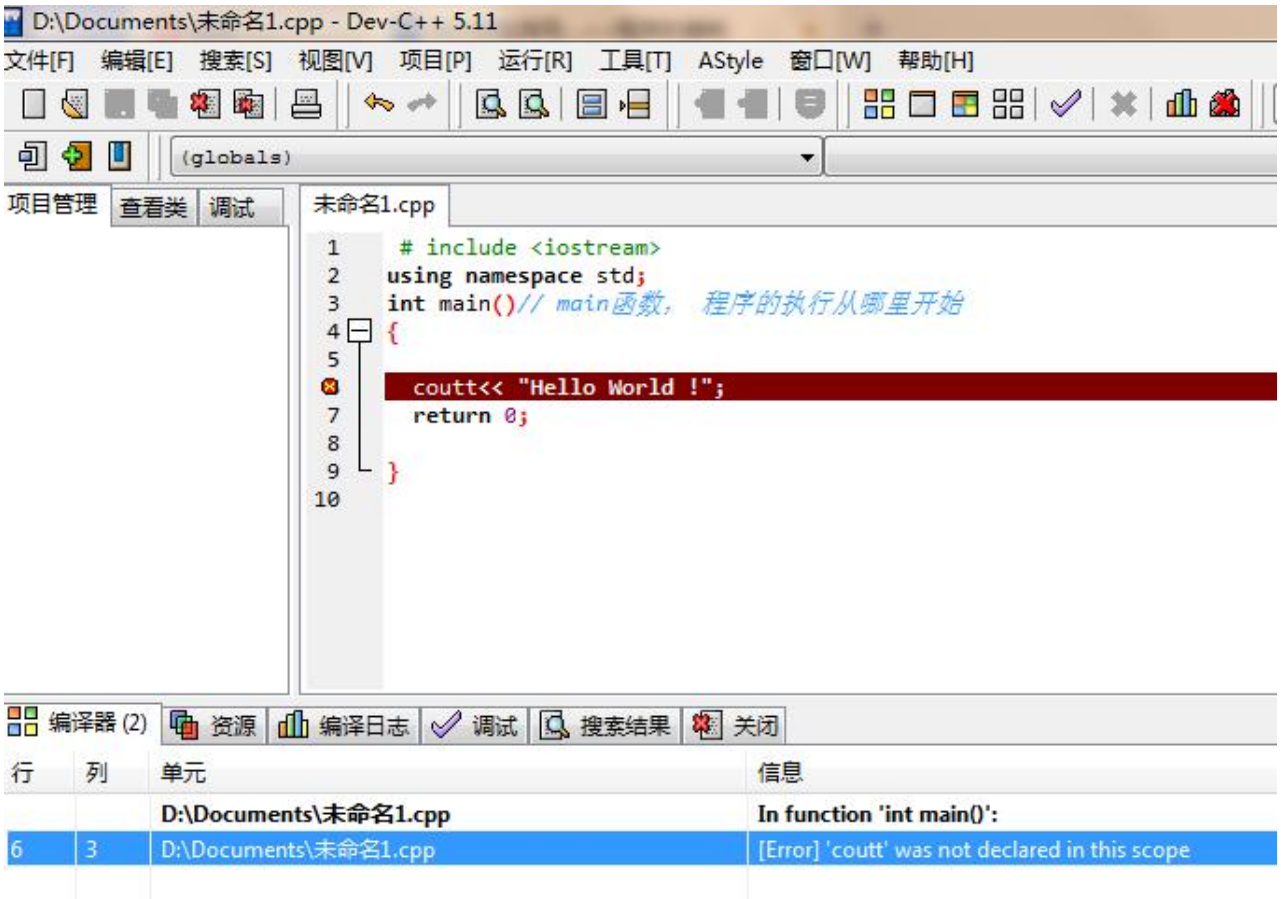
（6）return 0; ：这也是一个声明。该语句用于从函数返回值，并指示函数的完成。此语句基本上在函数中使用，以返回函数执行的操作的结果。

（7）缩进：如您所见，cout 和 return 语句已缩进或移到右侧。这样做是为了使代码更具可读性。在像 Hello World 这样的程序中，它似乎没有太大的相关性，但是随着程序变得越来越复杂，它使代码更易读，更不会出错。因此，您必须始终使用缩进和注释来使代码更具可读性。

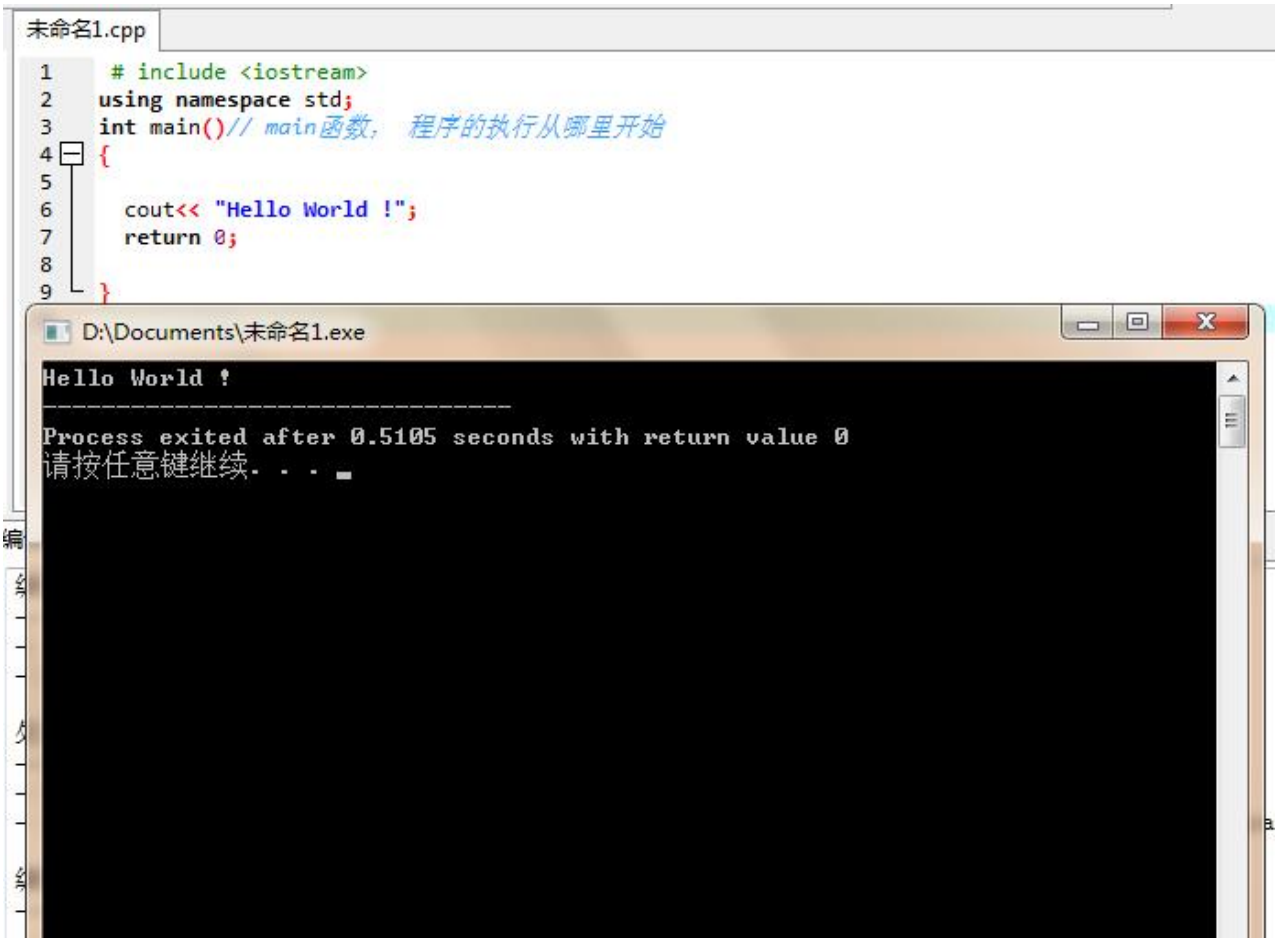
3、点击工具栏“运行”选择“编译 F9”—“运行 F10”或直接点“编译运行 F11”。



4、如果代码有错误，点击编译之后，编译器窗口会显示错误信息并且代码窗口有错误的行背景颜色标注。我们需要改正错误后重新编译——运行。



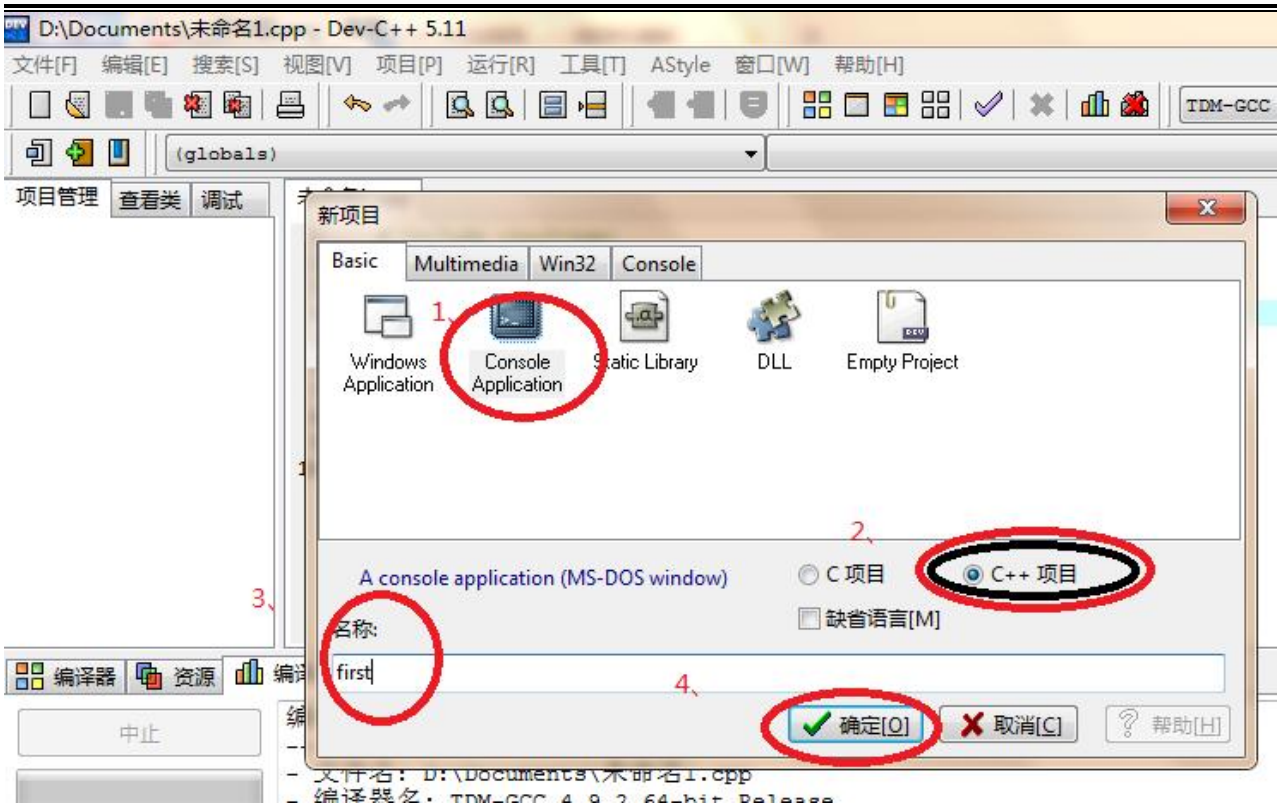
5、正确程序会在控制台输出一行“Hello World ！”。



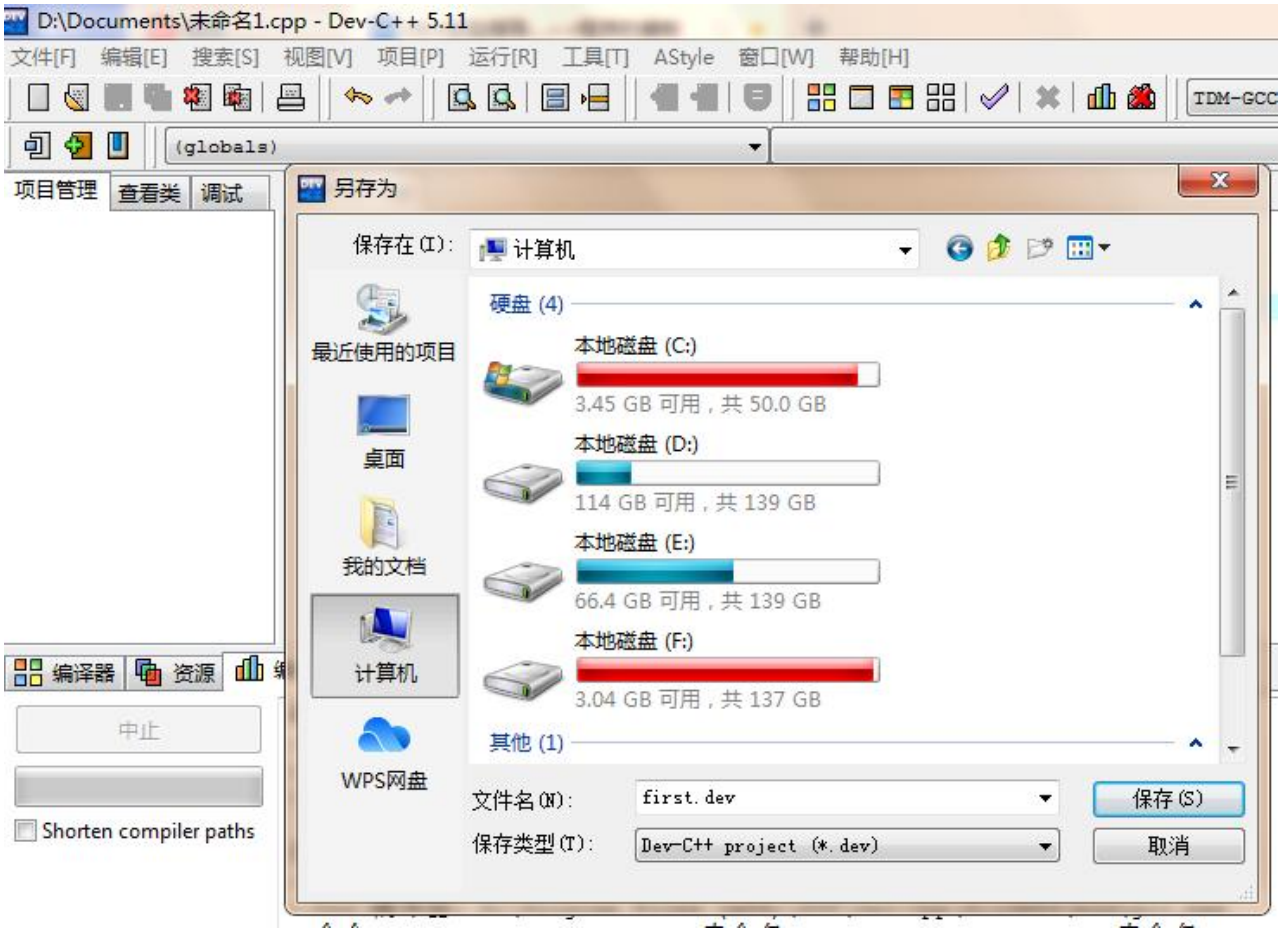
1.4.1.2 新建一个项目步骤如下：

1、打开 Dev-C++，单击文件--新建--项目后，会出现一个对话框。

2、选择第二个 console application(控制台程序)——>选择 C++项目——>输入项目的名称——>点击“确定”。

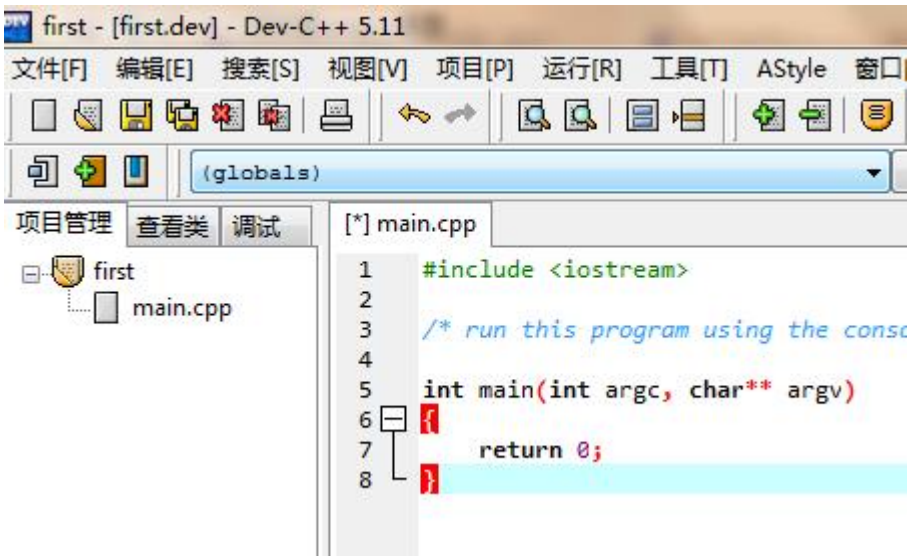


3、选择项目的保存路径。



4、保存完成之后，就进入程序的编辑了。可以看到 Dev-C++已经自动写好了一些代码，

这几行代码通常情况下都是必须的。



这里需要注意的：

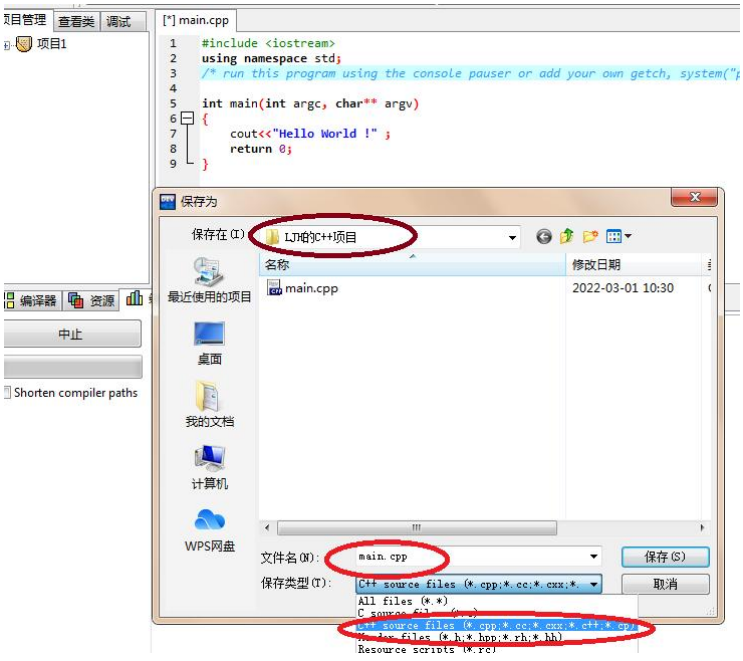
（1）main()括号内是固定的写法。

main(int argc,char **argv)，argv 为指针的指针，argc 为整数， char **argv or: char *argv[ ] or: char argv[ ][ ]。感兴趣的同学参看 https://zhuanlan.zhihu.com/p/140534864。

（2）目前阶段我们直接写 main()即可。

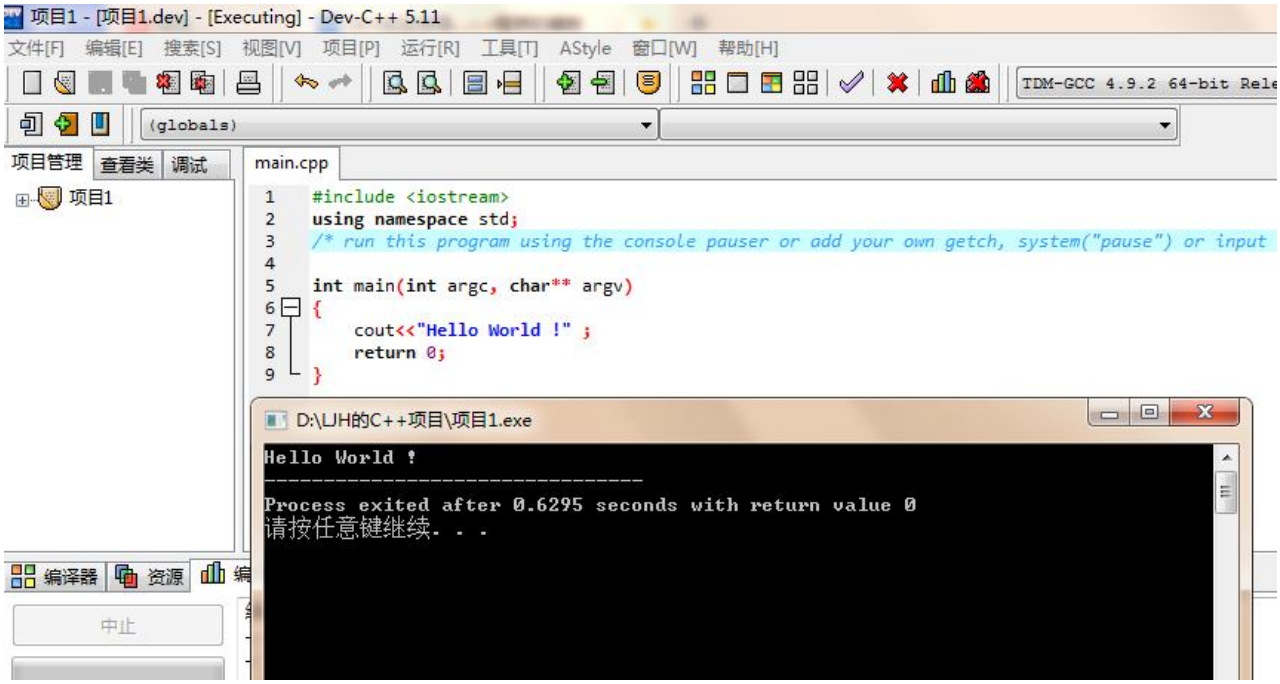
DEV c++里 void main（）个错误的写法，dev-c++比较严格，所以不接受 void main。VC 和 TC 上就可以通过。最正确的写法是 int main(void)。int main(void)这是 ANSI C 的标准，因为 DEV C++支持的是 ANSI C 标准。

5、输入几句简单的代码，把不需要的地方删掉。单击编译，看看程序有没有错。编译的时候，提示要把这个程序先保存。



保存时，需要选择存储路径，文件名字，文件类型选择“c++ source files”。

6、编译没有问题后，就可以执行了。



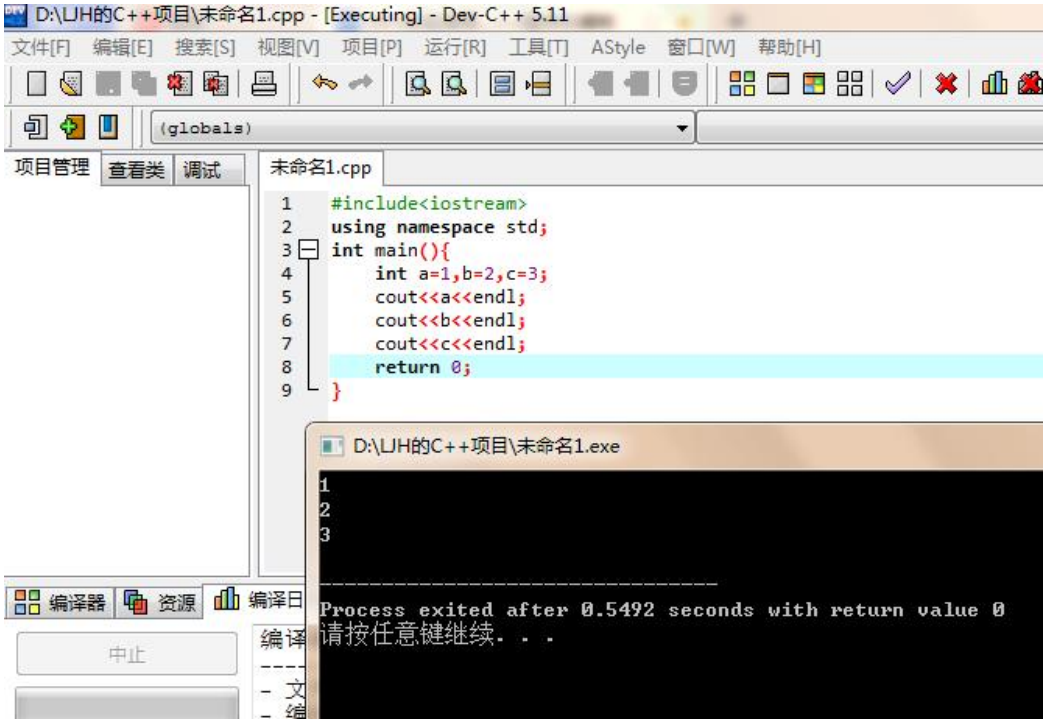
注意：需要添加 using namespace std。

OK 了，接下来你可以通过更改这个程序去练习 C++教材上面的程序了。

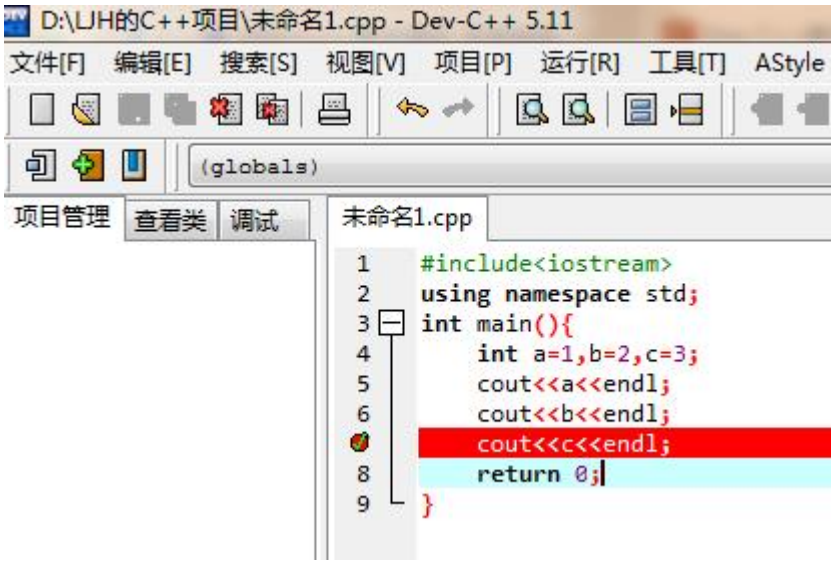
1.4.2 断点调试

没有比断点更常用的了，通过点击代码左边边栏就可以添加断点。值得注意的是并非每一行都可以添加断点，这个就留给大家去实践中体会吧。

1、写一段程序分三行输出 1、2、3。



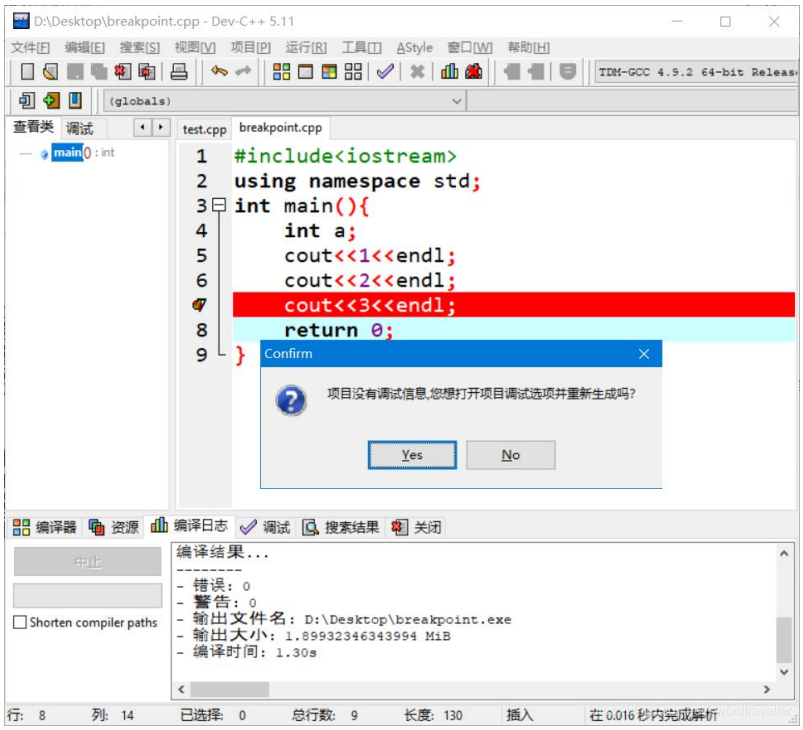
2、点击行号标红添加断点。



点击行号 7，则第 7 行就被标红，产生一个位于第 6 行和第 7 行之间的断点。

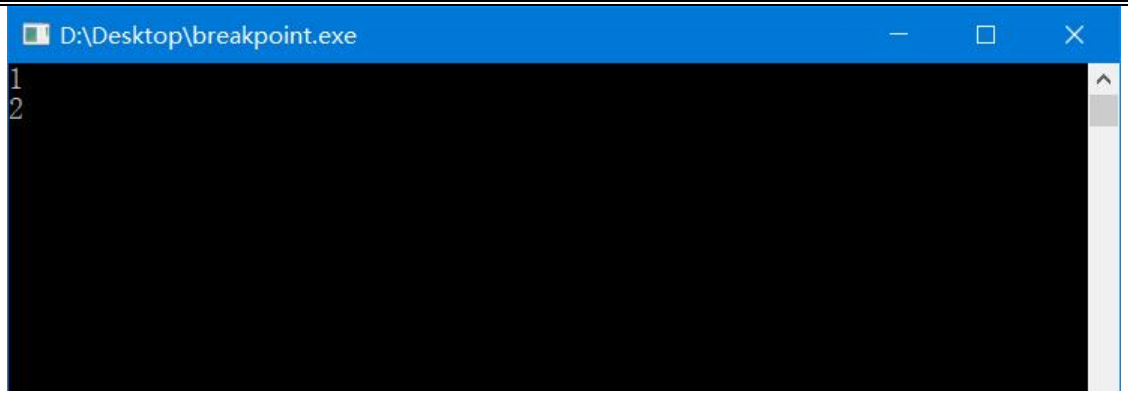
这一操作使得第 7 行之前产生了一个断点，当执行断点调试时程序只会执行 1-6 行，不会执行第 7 行及以后的内容。

3、F5 断点调试以及调试功能的配置。



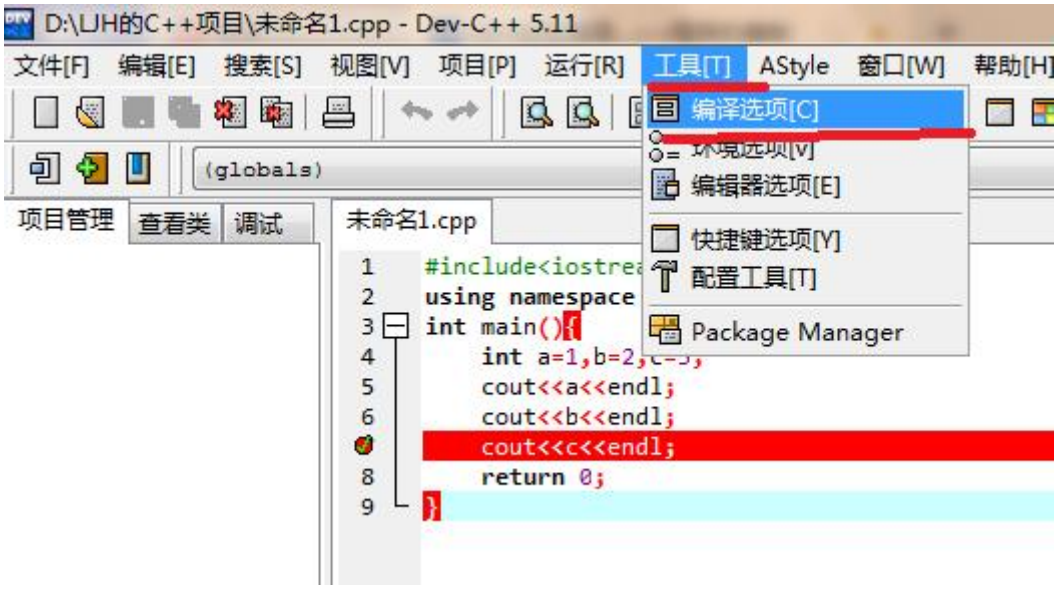
如果您是第一次用 Dev 断点调试，可能会遇到弹窗警告，这时点击 No 即可看到程序的断点调试命令窗口

点击 No，在弹出的命令行窗口中可以看到：

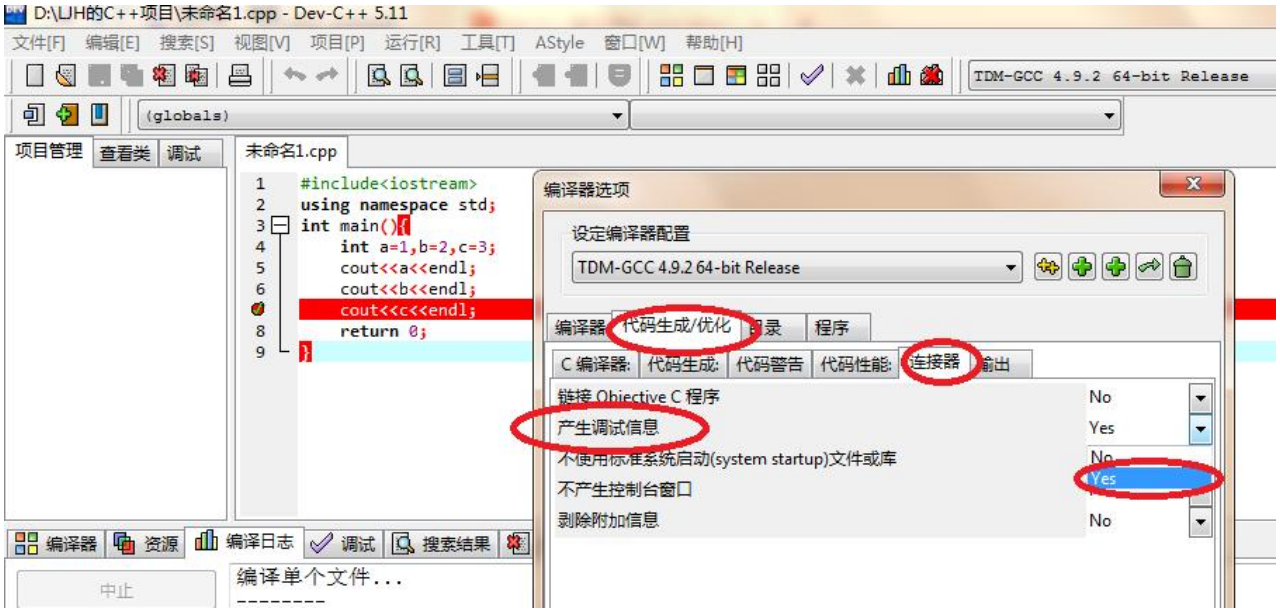


显然， 行 5 和行 6 都被执行了所以屏幕上输出了 1 和 2，但是没有 3，因为程序在行 7 之前中断了。

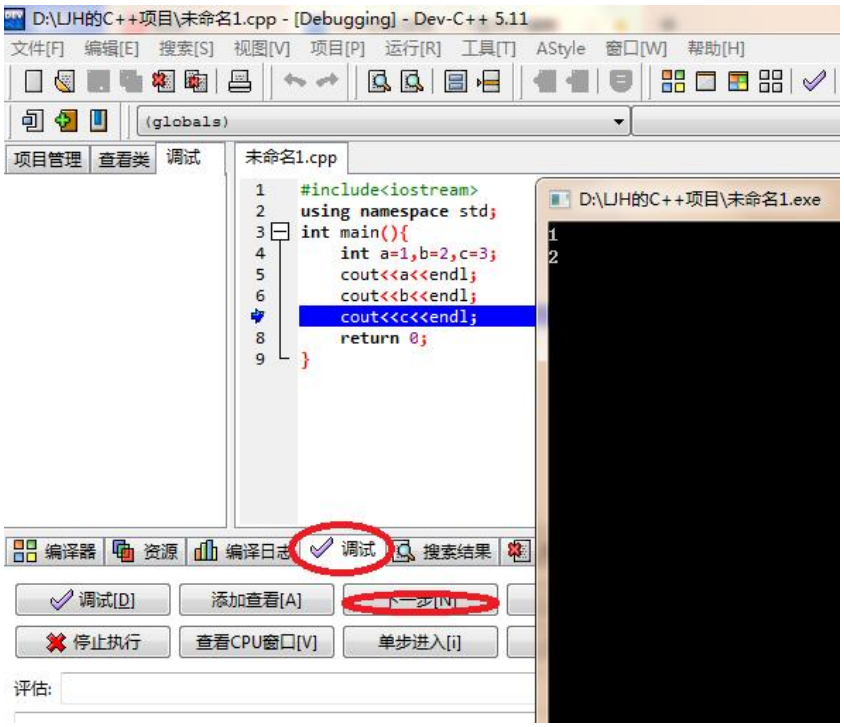
为了不被弹窗警告，关闭黑窗，配置如下：



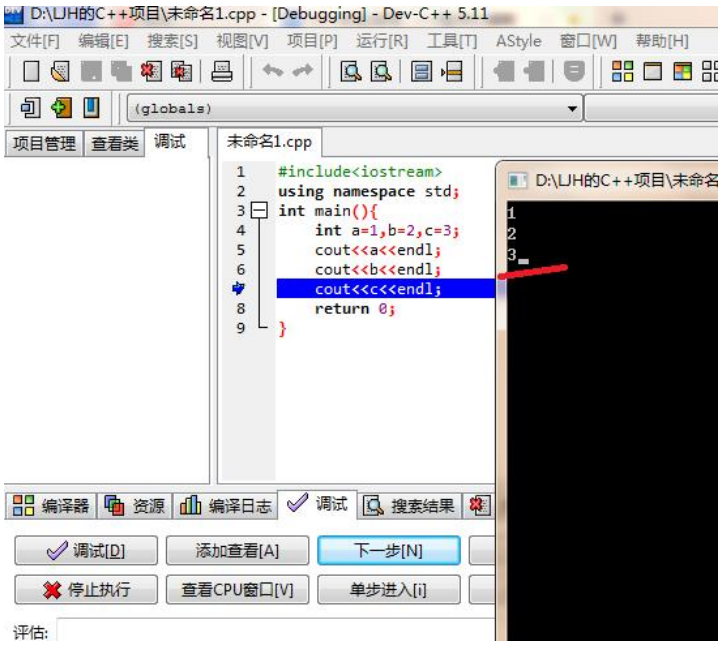
工具-编译选项- 代码生成/优化-连接器-产生调试信息，设为 Yes。



现在再按 F5：

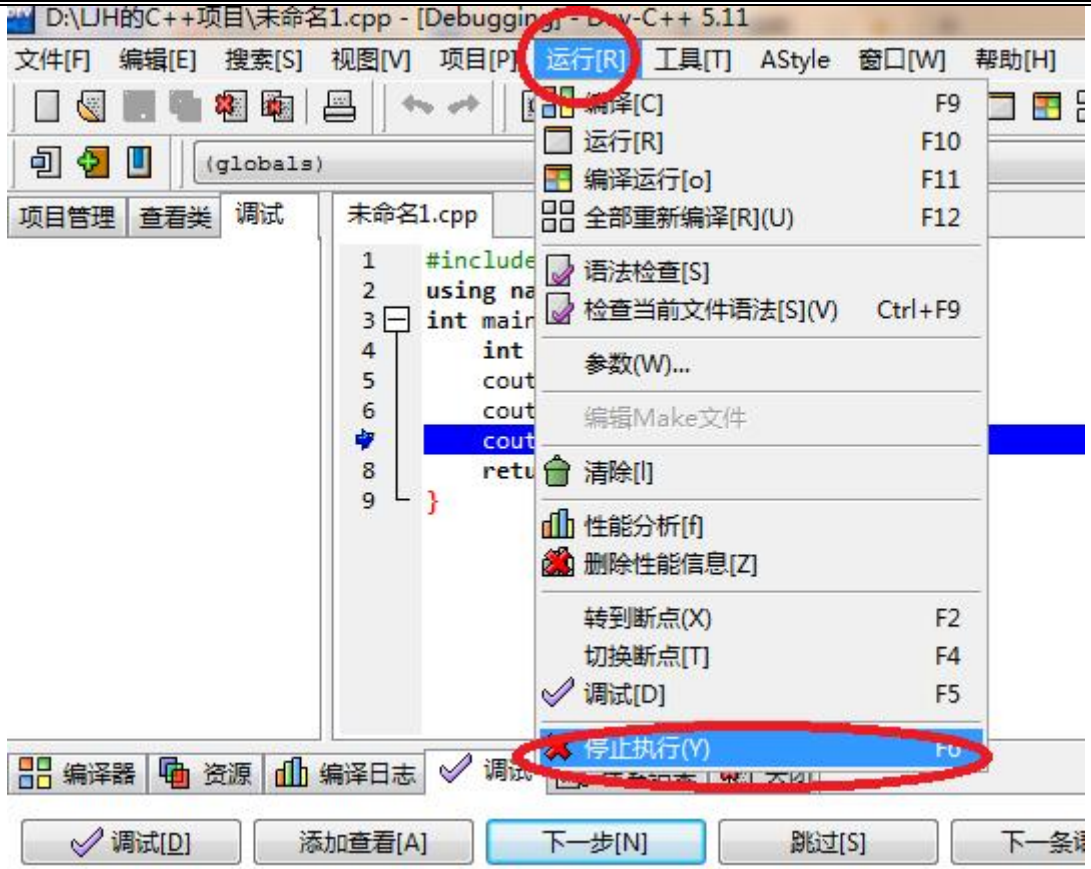


点击调试窗口“下一步”（或者按下 F7）



可以看到点了下一步之后，行 7 被执行了。

不要直接×掉黑窗，这样系统可能会稍微卡住一下，应该点工具栏“运行”中的“停止执行”。



我们这样就可以利用断点控制程序执行，调试程序。

1.4.3 实验任务

(1)运行下面程序，观察其输出，体会 i++与++i 的差别。

```
#include <iostream>
using namespace std;
int main()
{
    int myage=20,yourage=30;
    cout<<"I am"<<myage;
    cout<<"You are "<<yourage;
    myage++;
    ++yourage;
    cout<<"One year passes\n";
    cout<<"I am"<<myage;
    cout<<"You are"<<yourage;
    cout<<"Another year passes\n";
    cout<<"I am"<<myage++;
    cout<<"You are "<<++yourage;
    cout<<"I am"<<myage;
    cout<<"You are"<<yourage;
    return 0; }
```

(2)通过下面程序验证你所使用系统上运行的C++编译器中每个基本数据类型的长度。

```
#include <iostream>
```

```
using namespace std;
int main()
{
    cout << "char length:" << sizeof( char ) << endl;
    cout << "int length:" << sizeof( int ) << endl;
    cout << "long length:" << sizeof( long ) << endl;
    cout << "float length:" << sizeof( float ) << endl;
    cout << "double length:" << sizeof( double ) << endl;
    cout << "long double length:" << sizeof( long double ) << endl;
    cout << "char* length:" << sizeof( char* ) << endl;
    cout << "int* length:" << sizeof( int* ) << endl;
    cout << "long* length:" << sizeof( long* ) << endl;
    cout << "float* length:" << sizeof( float* ) << endl;
    cout << "double* length:" << sizeof( double* ) << endl;
    cout << "long double* length:" << sizeof( long double* ) << endl;
    cout << "void* length:" << sizeof( void* ) << endl;
    return 0;
}
```

(3)掌握各种运算符。

假设有变量说明：

```
char c1='a', c2='B', c3='c';
int i1=10, i2=20, i3=30;
double d1=0.1, d2=0.2, d3=0.4;
```

先写出下列表达式的值，然后上机验证。

(a) c1+i2*i3/i2%i1;　　(b) i1++ +i2%i3;

(c) i2-- * ++i3;　　(d) i1>i2>i3<d1<d2<d3;

(e) (c1=i2*i3)!=(i2%i1)　　(f) d1>d2||i1==i2;

(g) c1>i1?i1:c2;　　(h) 0?1:2?0:0?3:4;

(i) i1+=i2*=i3;　　(j) i3=(i1=1,i2--);

(k) i1=(c1,c2,c3);　　(l) !i1&&i2--;

可以使用下面程序框架上机验证。

```
#include <iostream>
using namespace std;
int main()
{
  char c1='a', c2='B', c3='c';
  int i1=10, i2=20, i3=30;
  double d1=0.1, d2=0.2, d3=0.4;
   数据类型说明符 x;
  x= (表达式);
```

```
    cout<<"x="<<x<<endl;
    return 0;
}
```

（4）功能需求：运行时显示"Menu: A(dd) D(elete) S(ort) Q(uit)， Select one:"提示用户输入，A 表示增加，D 表示删除，S 表示排序，Q 表示退出，输入为 A、 D、S 时分别提示"数据已经增加、删除、排序。"输入为 Q 时程序结束。

按照上述功能需求写两个程序，分别使用if分支语句和switch分支语句实现：

程序1要求:使用 if … else 语句进行判断，用 break、continue 控制程序流程。

程序2要求:使用 Switch 语句实现。

(5)找出2~10000之内的所有完全数。所谓完全数，即其各因子之和正好等于本身的数。如6=1+2+3，28=1+2+4+7+14，所以6，28都是完全数。