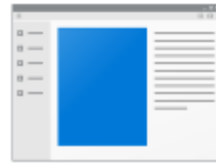




源程序和可执行程序



Hello.cpp



Hello.exe

- 源程序：C++ 语言代码，人来阅读。
- 可执行程序：机器代码，计算机来执行。

程序暂停，使用 `system("pause")`，必须引用 `cstdlib` 库。

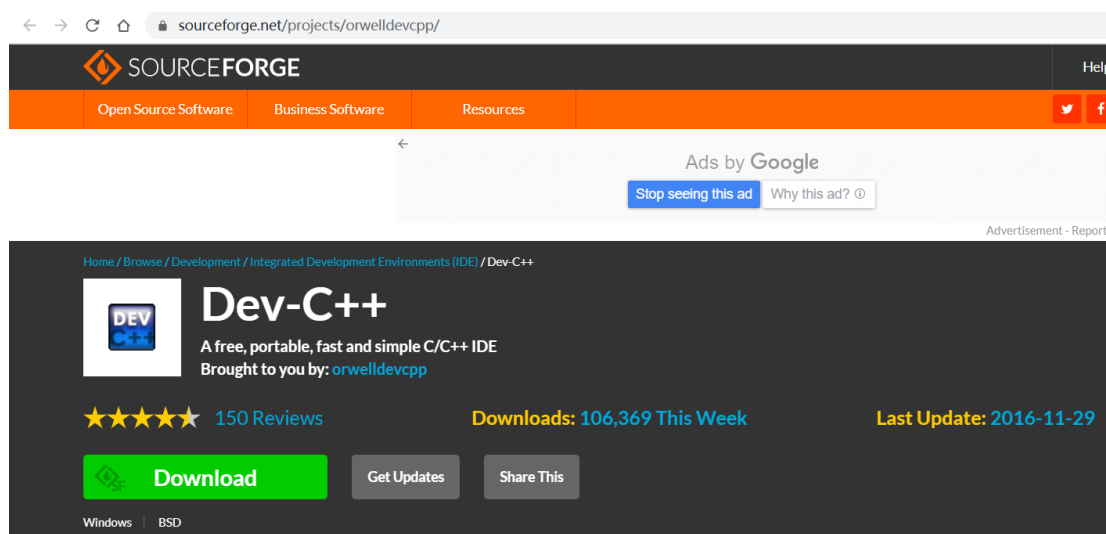
```
#include<iostream>
#include<cstdlib>
using namespace std;
int main()
{
    cout << "Hello, Xiaozhi!";
    system("pause");
    return 0;
}
```



C++开发工具的使用（DEV C++）

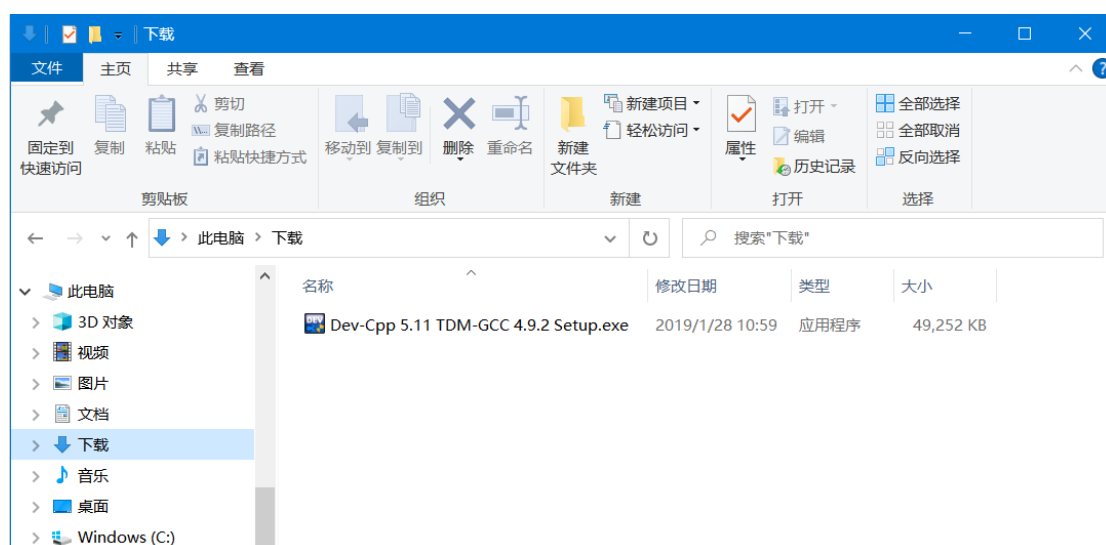
【下载】

- 打开浏览器，在地址栏输入 sourceforge.net/projects/orwelldevcpp/，在打开的页面里面点击 Download 图标，这时就会下载 Dev-C++ 的安装包。



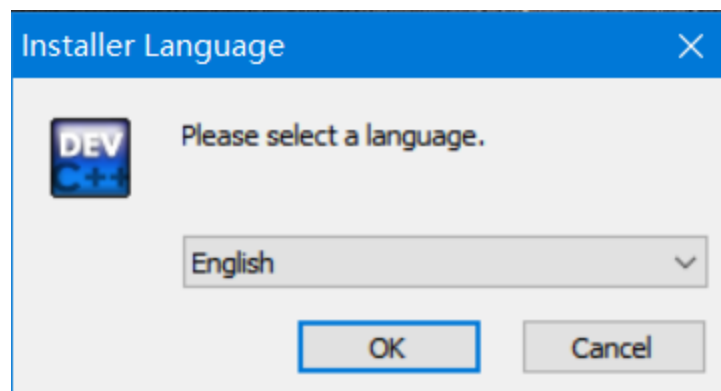
【安装】

- 在下载目录里，找到下载好的安装包，双击文件名，运行安装程序。

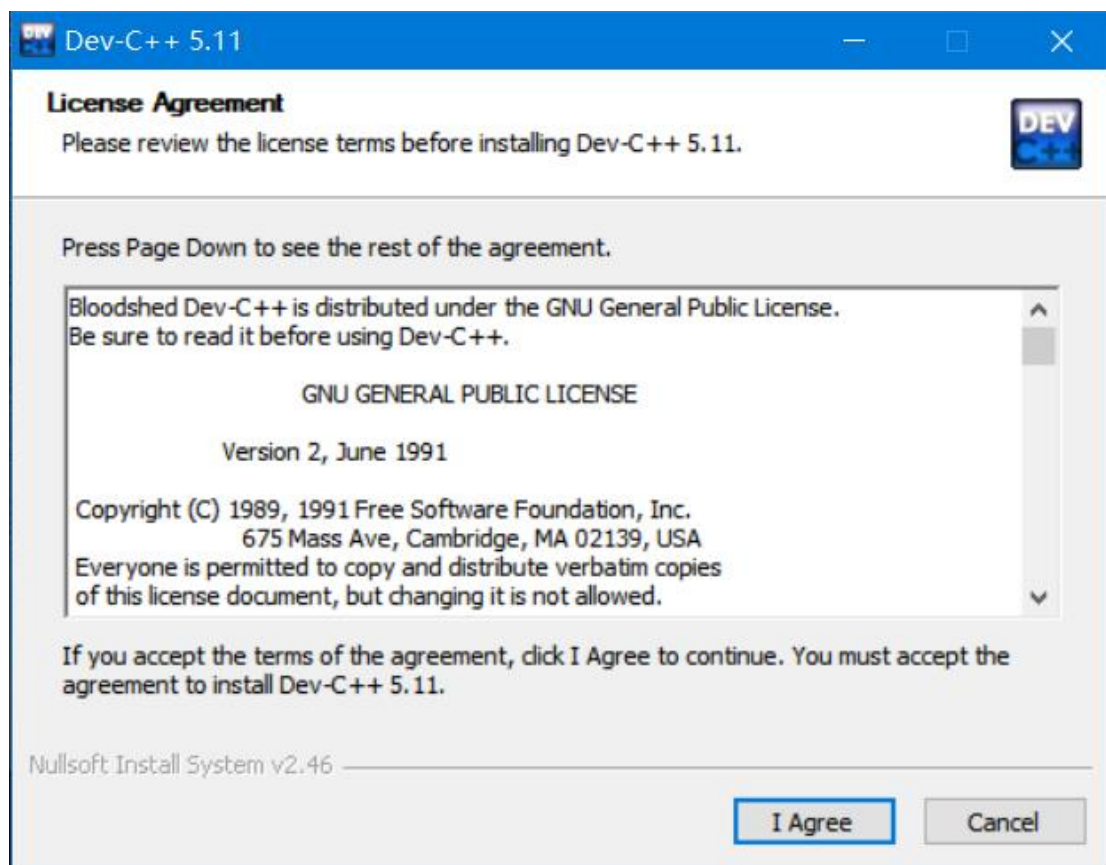




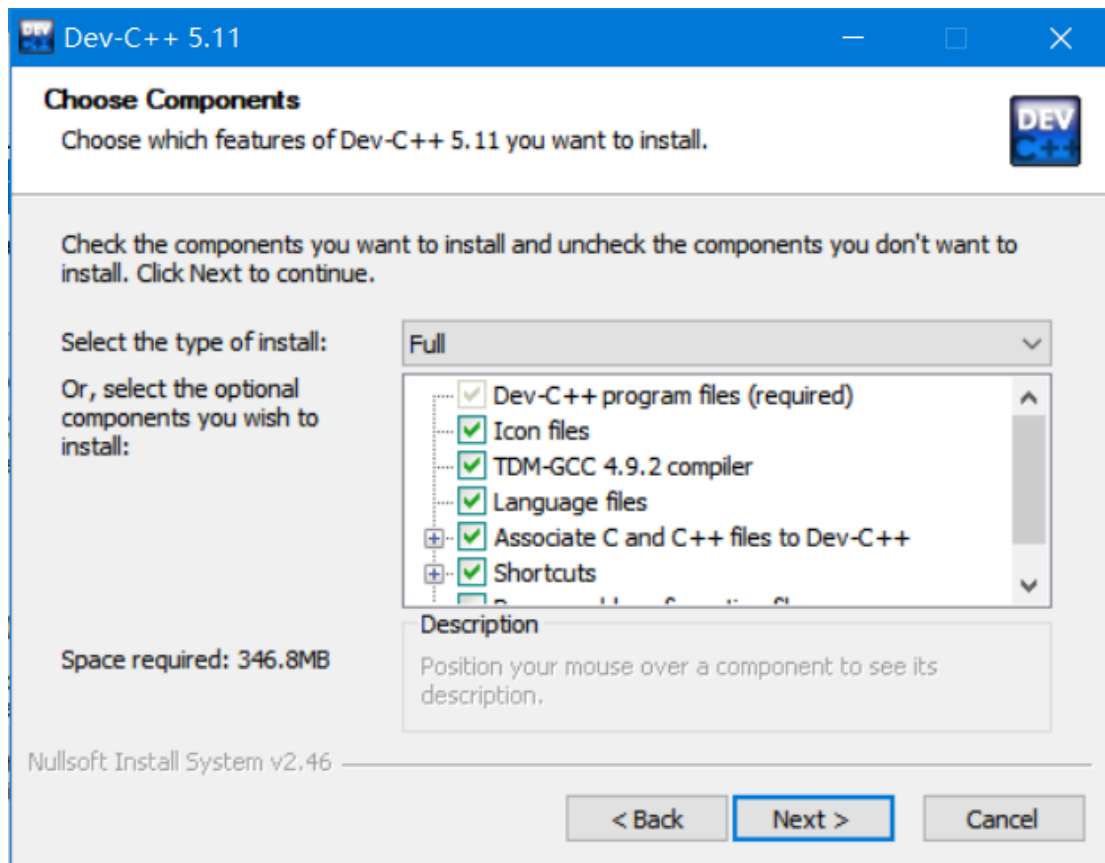
- 如果 Windows 弹出用户账号控制窗口，点击“是”按钮。
- 安装程序启动后，弹出选择语言的窗口，直接点击“OK”按钮。



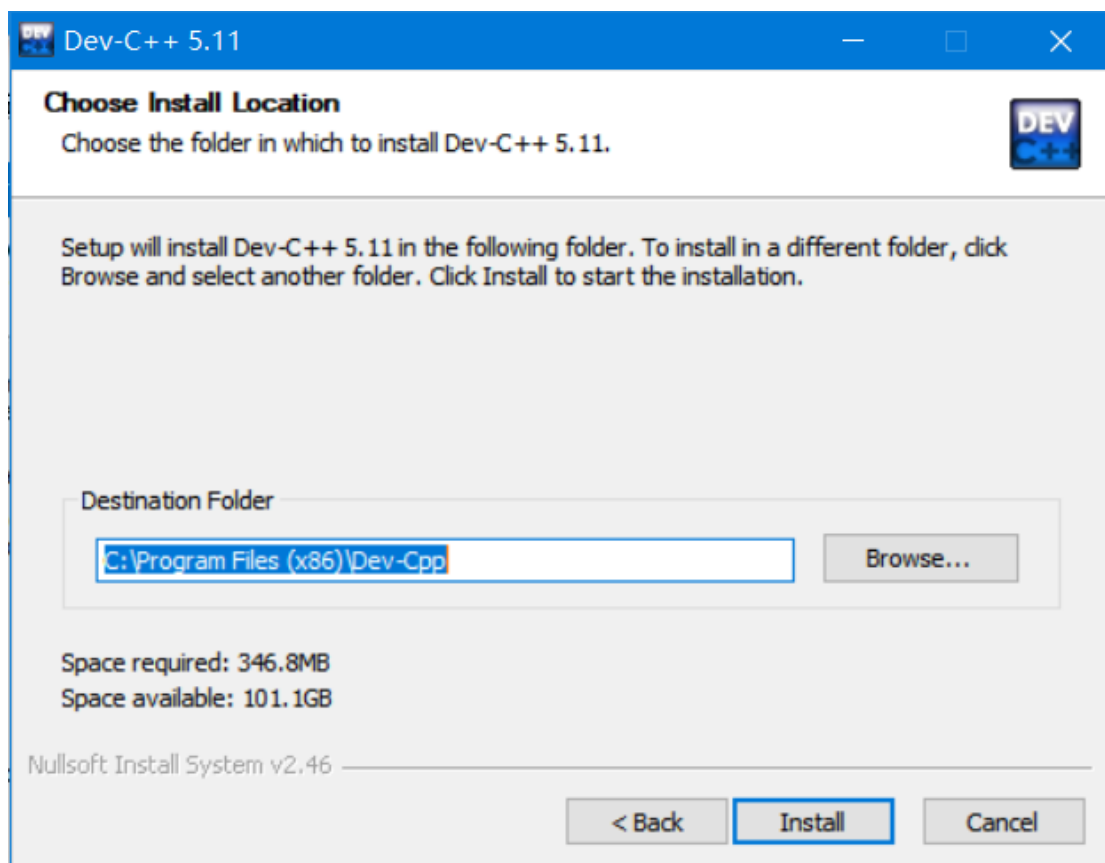
- 在许可协议窗口点击“I Agree”按钮。



- 在选择安装组件窗口，点击“Next>”按钮。

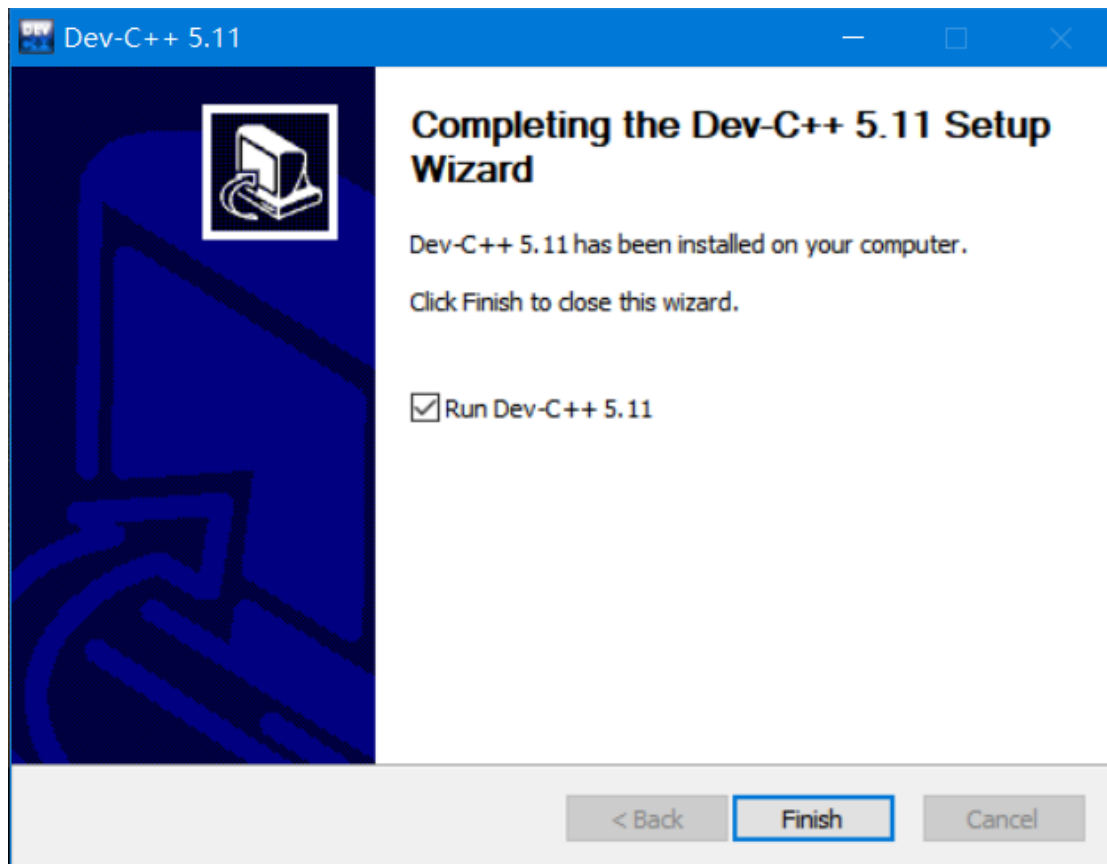


- 在选择安装路径窗口，点击“Install”按钮进行安装。



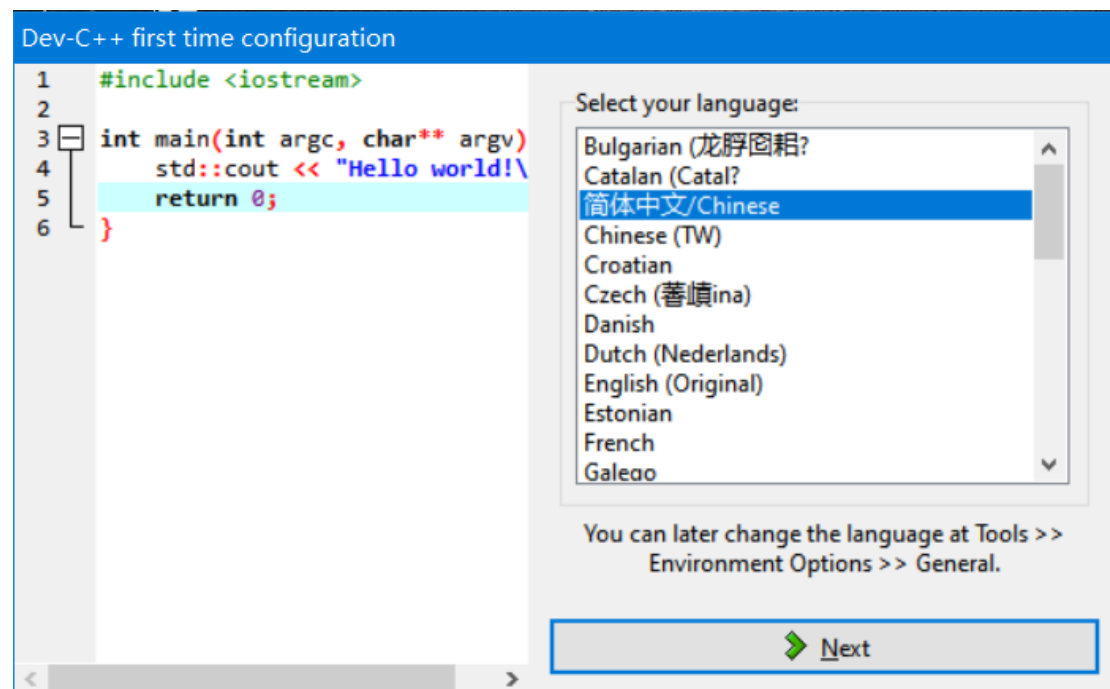


- 最后，当显示这个窗口时，表示安装成功，点击“Finish”按钮关闭窗口，并运行 Dev-C++。

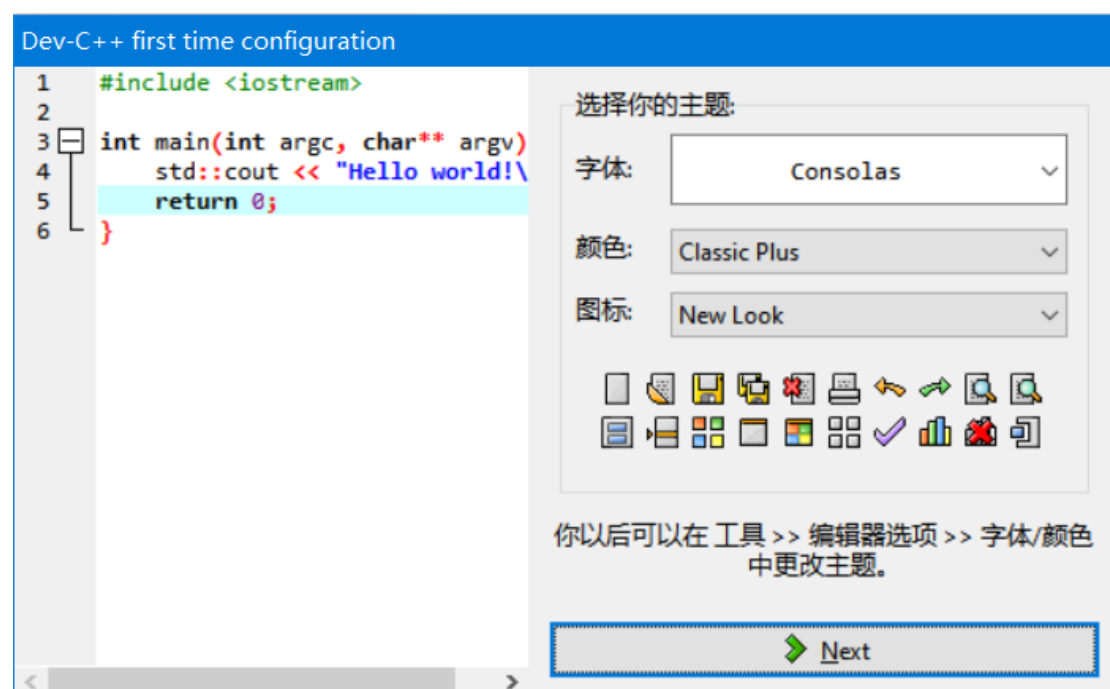


【第一次运行】

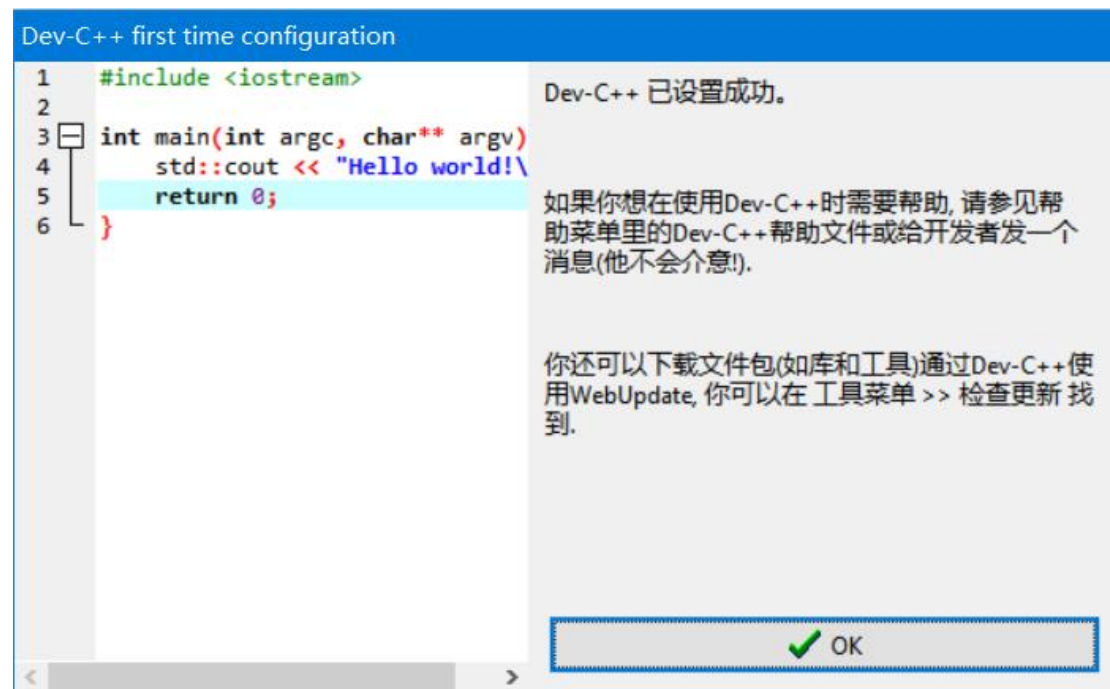
- 第一次运行 Dev C++ 时，会弹出自定义设置窗口。
- 可以选择简体中文为用户界面语言，然后点击“Next”按钮。



- 选择默认主题，或者选择自己喜欢的字体、颜色和图标样式，然后点击“Next”按钮。

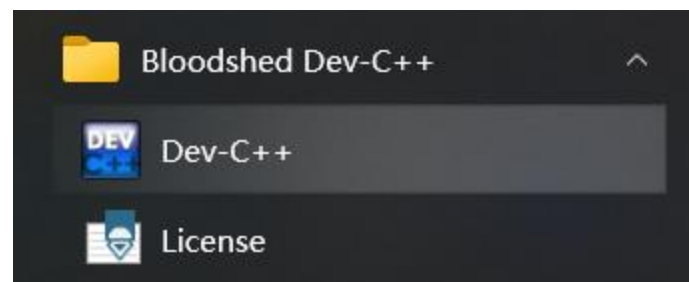


- 出现这个界面，表示设置成功，点击“OK”按钮。
- 接下来就可以使用 Dev C++，开始你的 C++ 编程之旅吧！



【启动】

- 方法一：点击开始按钮，从程序菜单里面找到“Bloodshed Dev-C++”文件夹，然后点击 Dev-C++ 程序图标。



- 方法二：双击桌面上的 Dev-C++ 的图标。



- 方法三：在搜索栏输入 Dev。然后点击搜索结果里面的 Dev-C++ 或按回车键。

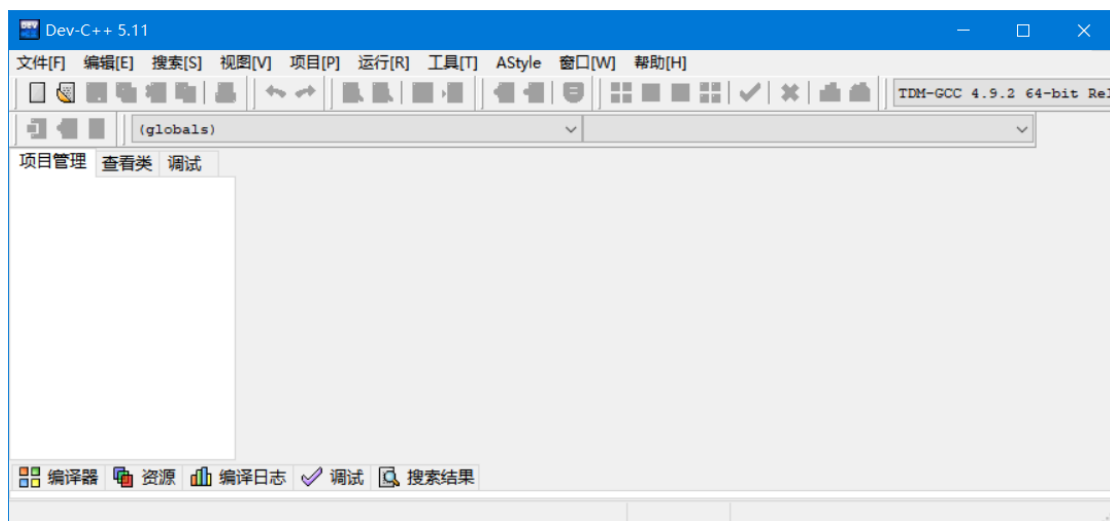




最佳匹配

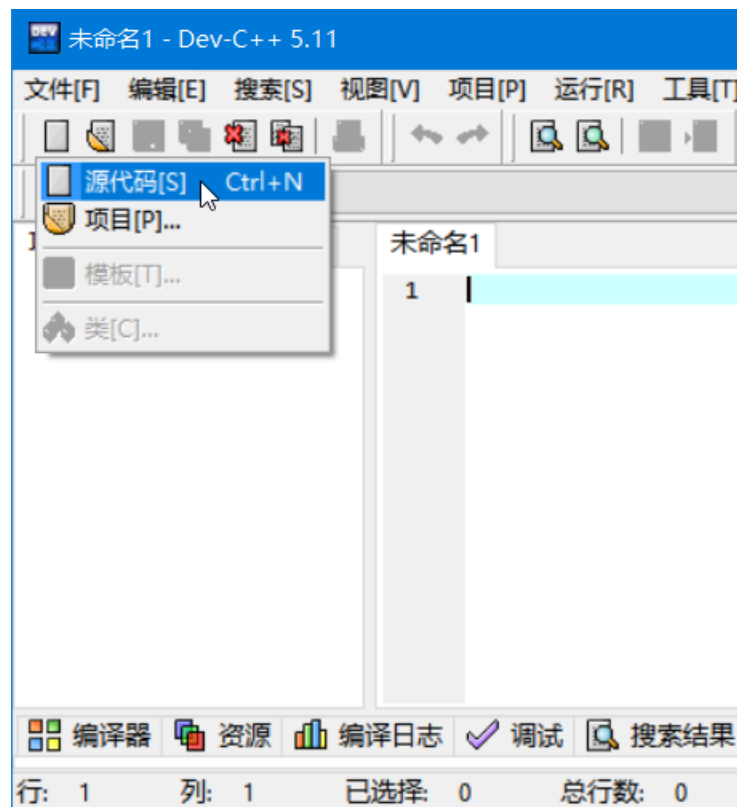


【主界面】

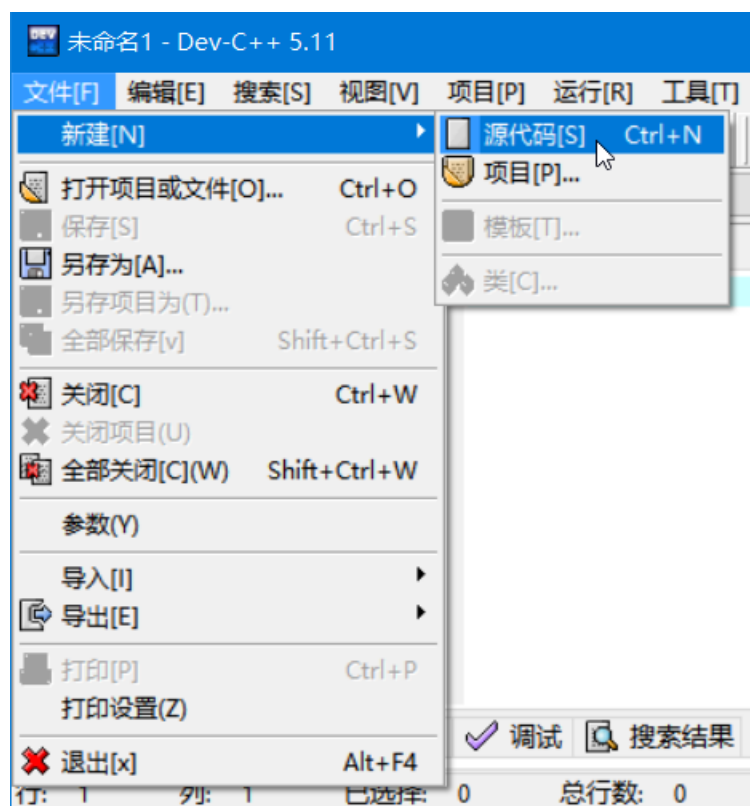


【新建源代码】

- 方法一：使用工具按钮。



- 方法二：使用菜单



- 方法三：使用快捷键 Ctrl + N。即按住 Ctrl 键不松，再按下 N 键。

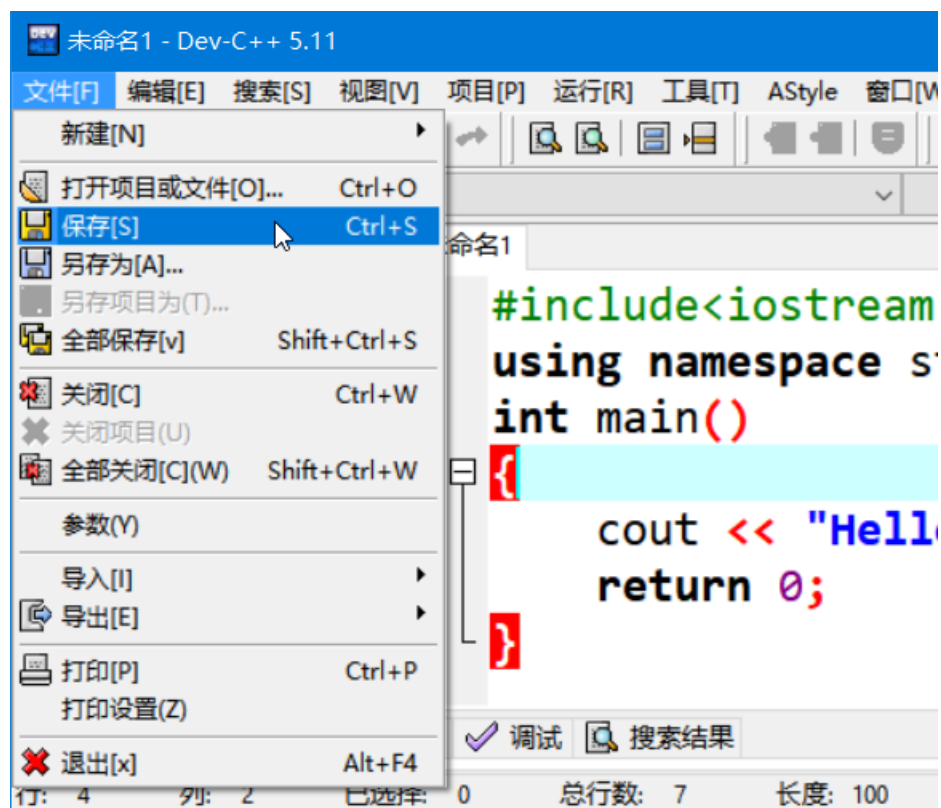
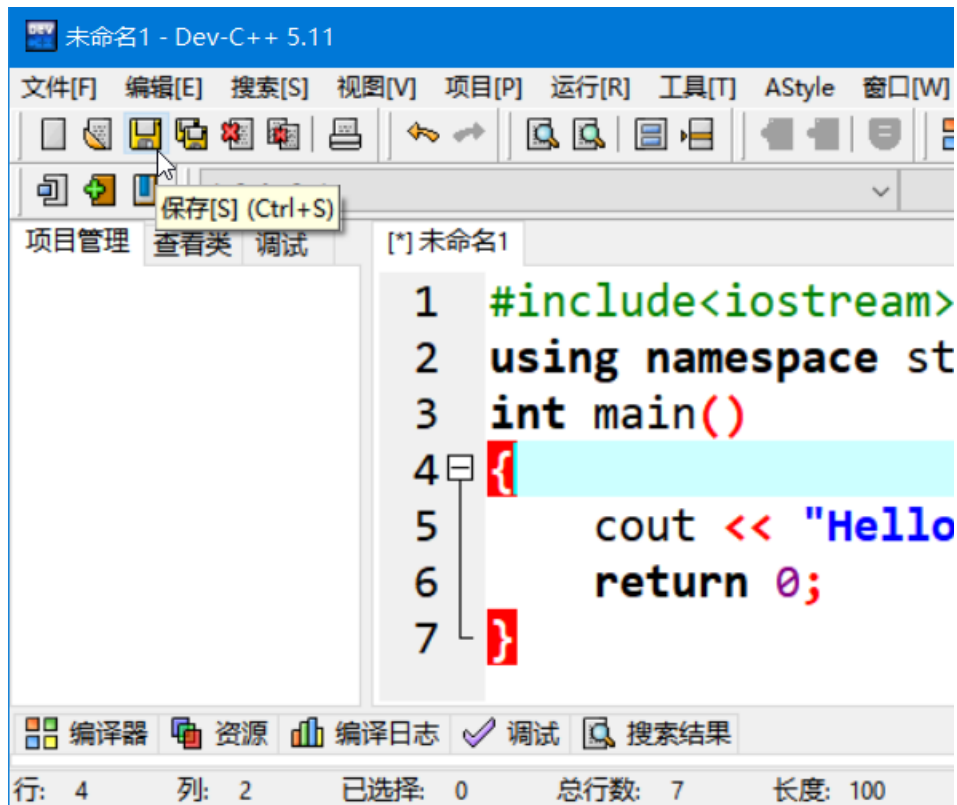


【编写源代码】

- 在代码编辑区输入源代码

【保存源代码】

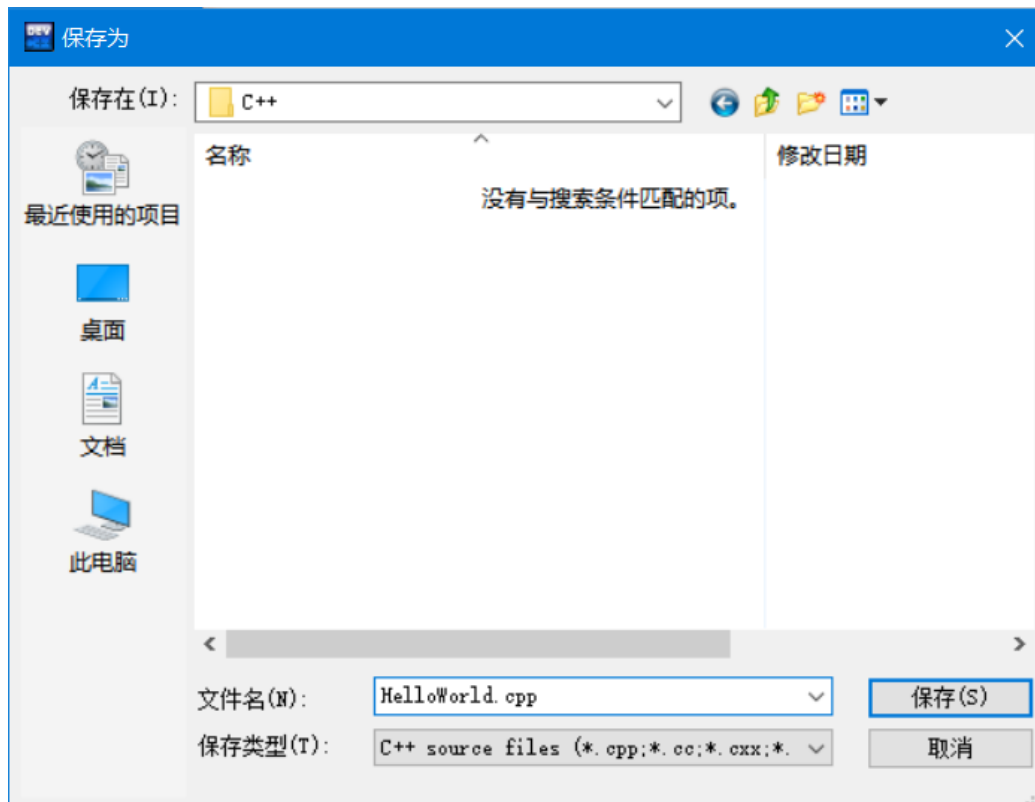
- 方法一：点击“保存”按钮。
- 方法二：点击“文件”菜单，再点击“保存”菜单项，或者“另存为”菜单项。
- 方法三：按下快捷键 Ctrl + S。



- 在弹出窗口中，选择一个目的文件夹，再给源代码起一个有意义的名字，点击“保存”按钮。

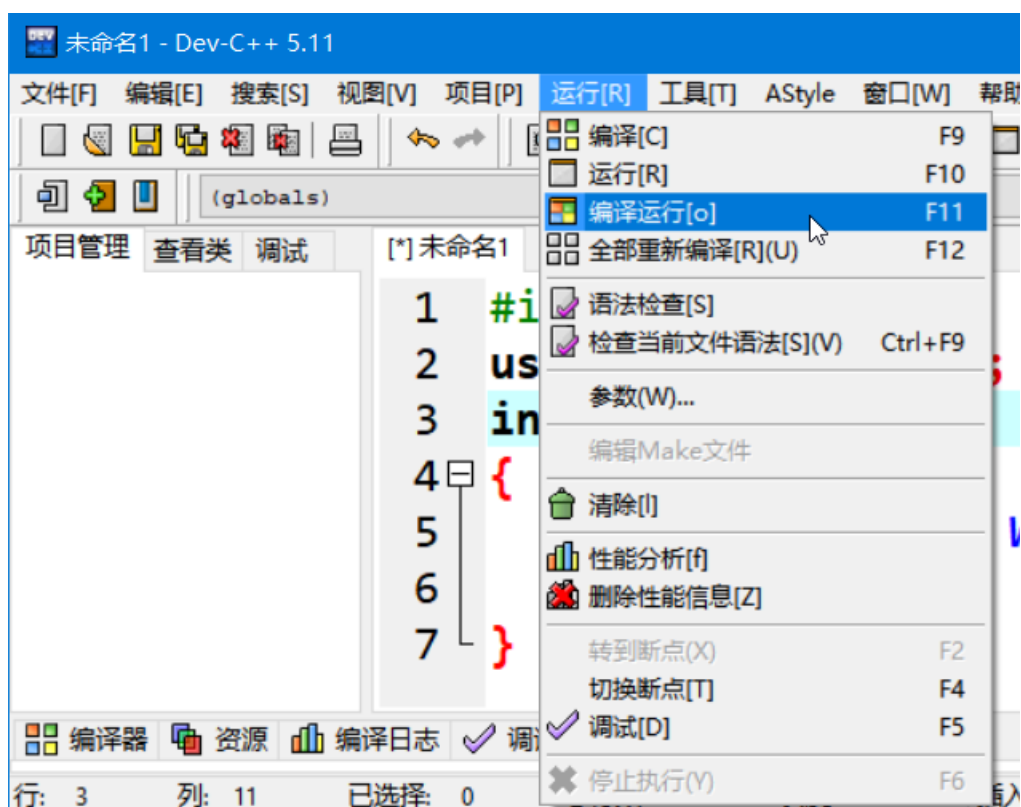
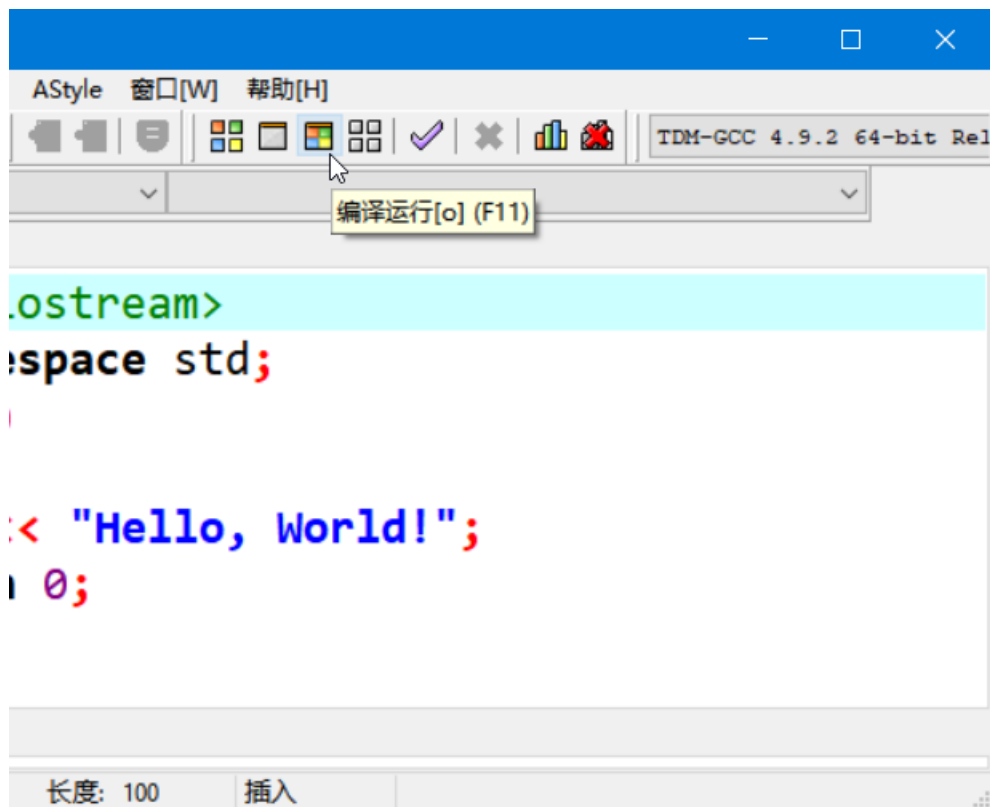


- 一个好的习惯是创建了一个新源代码后，在还未输入代码之前先将该程序保存到硬盘某个目录下，然后在编辑代码过程中经常性地保存程序，以防止机器突然断电或者死机。



【编译和运行】

- 方法一：点击“编译运行”按钮。
- 方法二：点击“运行”菜单，再点击“编译运行”菜单项。
- 方法三：按下快捷键 F11。



- 如果程序中存在词法、语法等错误，则编译过程失败。
- 编译器会在屏幕下方的“编译器”标签页中显示红色的错误信息。



编译器			资源	编译日志	调试	搜索结果	关闭
行	列	单元	信息				
		D:\学习文件夹\C++\HelloWorld.cpp	In function 'int main()':				
5	29	D:\学习文件夹\C++\HelloWorld.cpp	[Error] 'endl' was not declared in this scope				

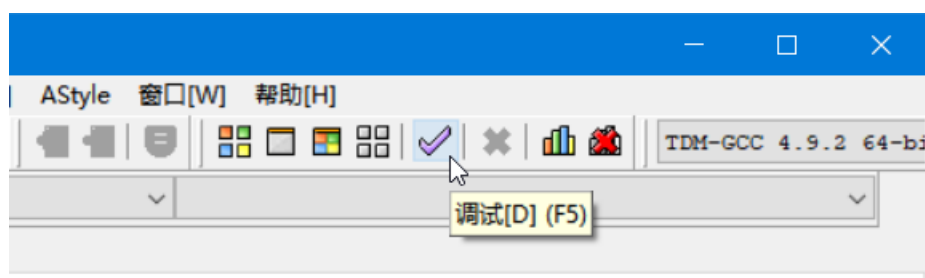
- 双击这条错误信息，Dev C++会将源程序相应的错误行标成红色底色。

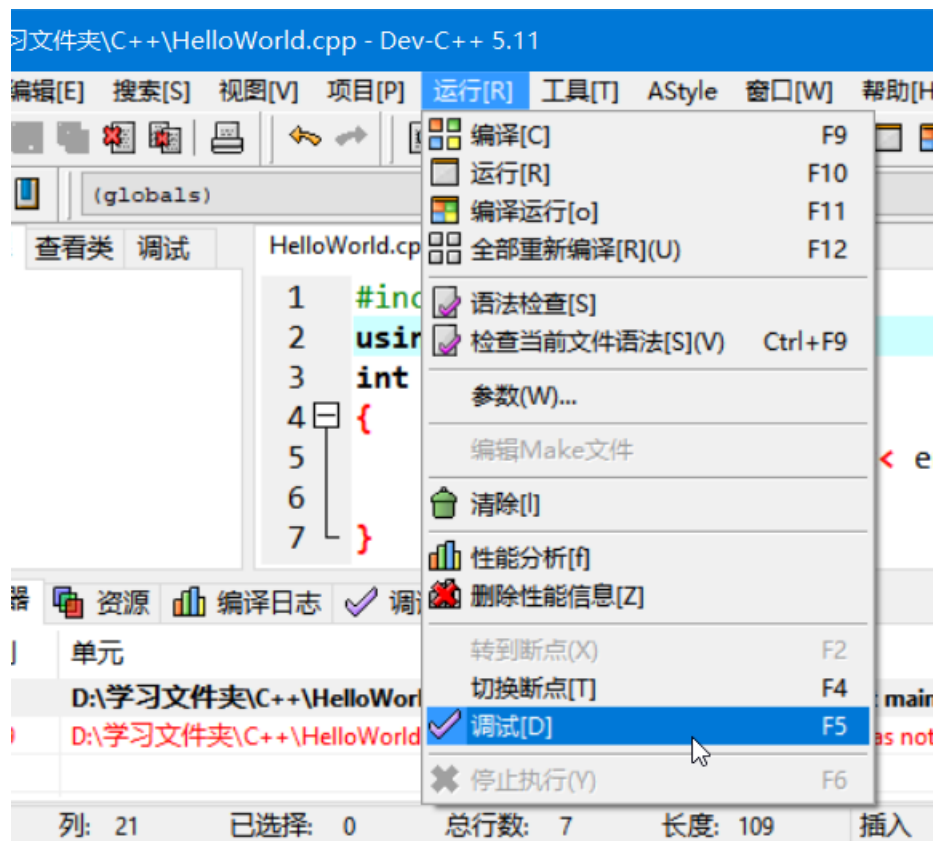
```
HelloWorld.cpp
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      cout << "Hello, World!" << endl;
6      return 0;
7  }
```

- 通过预处理、编译和链接的程序无法发现程序深层次的问题（比如算法不对导致结果不正确）。

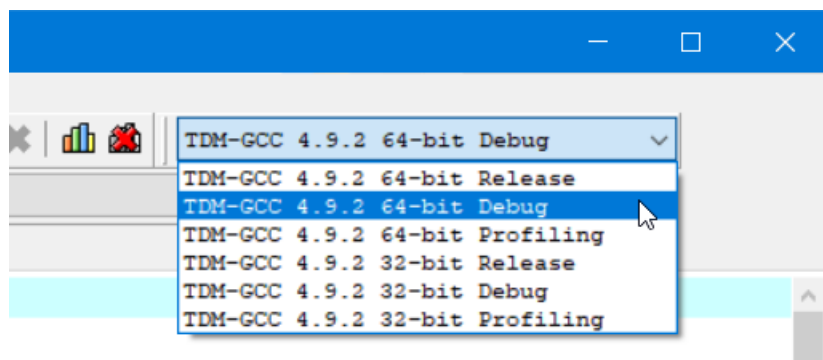
【调试程序】

- 方法一：点击“调试”按钮。
- 方法二：点击“运行”菜单，再点击“调试”菜单项。
- 方法三：按下快捷键 F5。

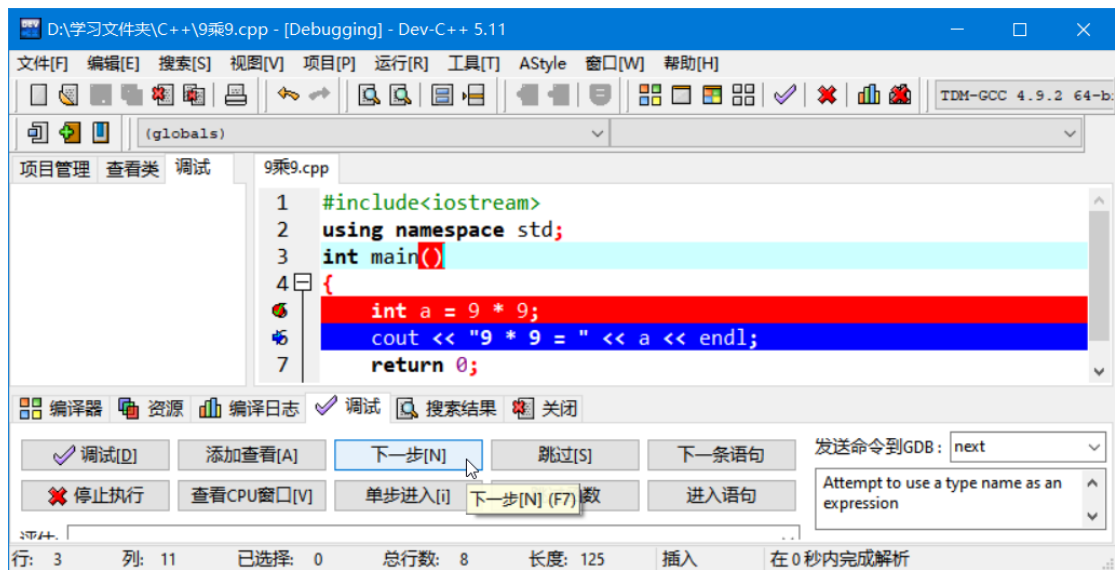




- 注意：编译器只有配置为调试模式才能进行代码调试。
- 可以选择 64-bit Debug 或者 32-bit Debug。



- 深层次的问题，可以采用单步执行（Debug）的方式调试程序。
- 点击“下一步”按钮，可以让程序单步执行。
- 如果要停止调试，可以点击“停止执行”按钮。



- 设置程序断点

- ✓ 如果想让程序运行到某一行前能暂停下来,就需要将该行设成断点。

具体方法是在代码所在行行号上单击,该行将被加亮。

- ✓ 如果想取消不让某行代码成为断点,则在代码行号再次点击即可。

- 观察程序变量的值

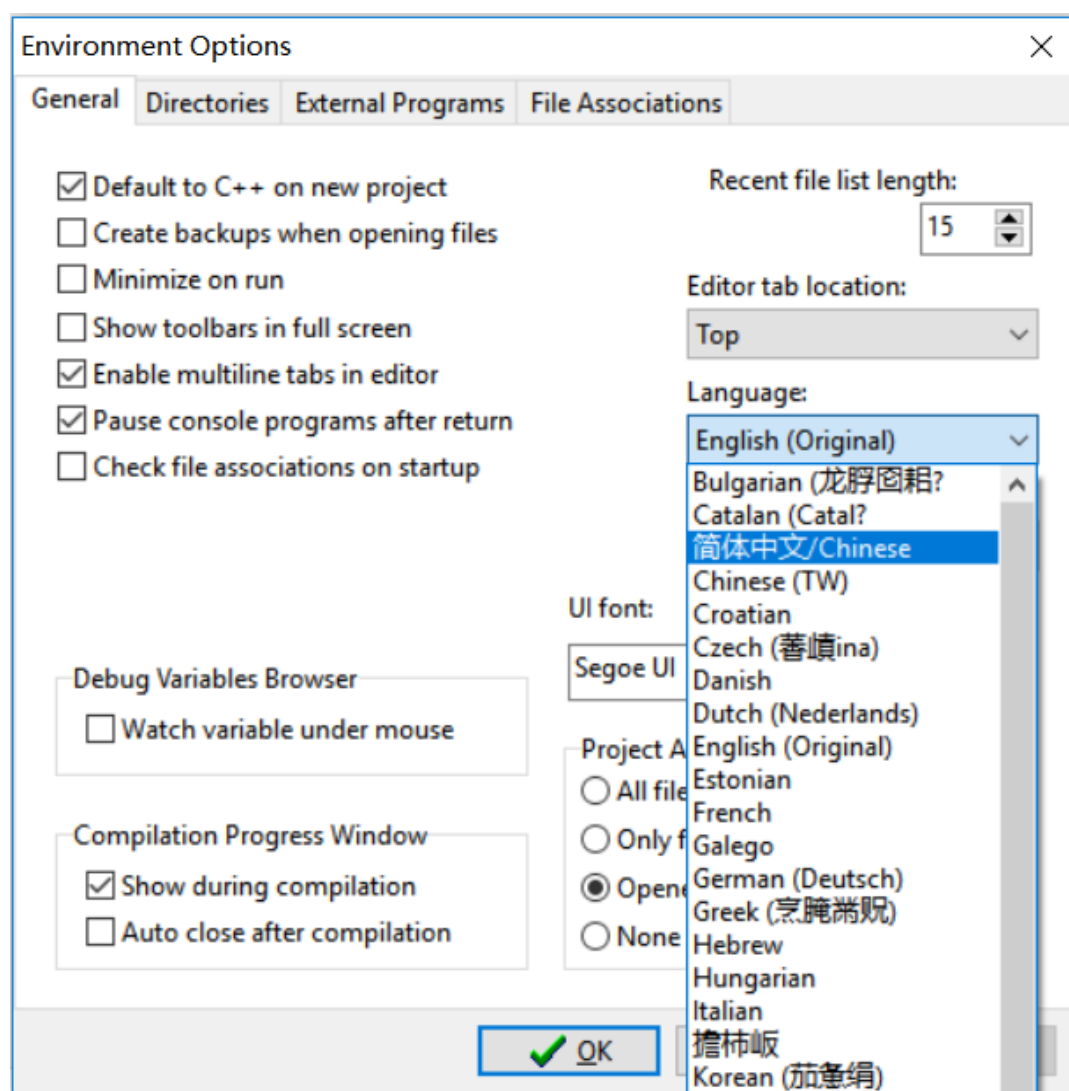
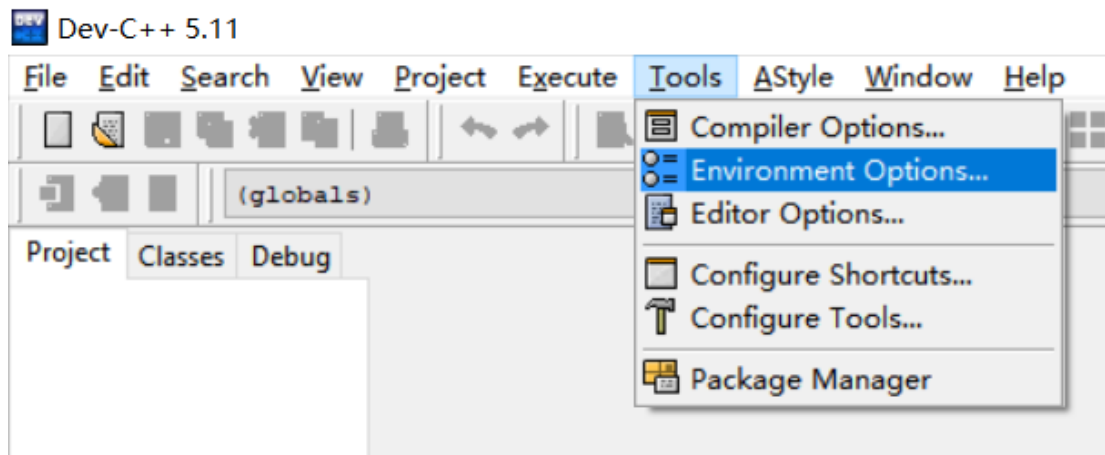
- ✓ 选中需要观察的变量,右键点击“添加查看”菜单项。

- ✓ 即可在左边调试窗口,看到变量的值。



【改变界面语言】

- 步骤一: 点击“Tools”菜单。
- 步骤二: 点击“Environment Options”菜单
- 步骤三: 在 Language 下拉列表里选择简体中文。

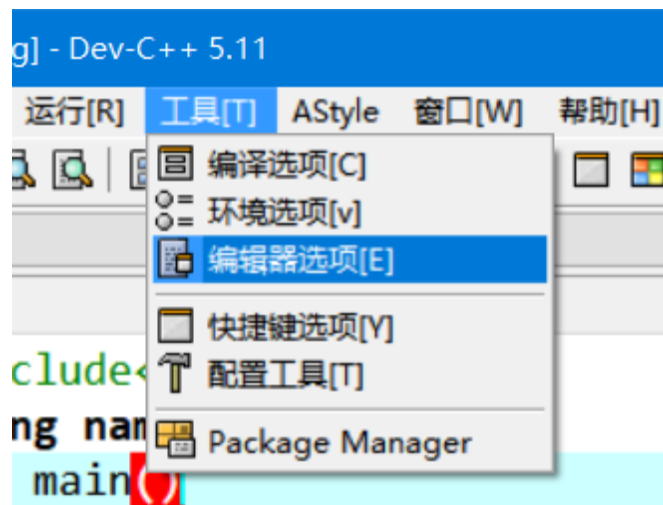


【改变编辑器字体大小】

- 方法一：按照 Ctrl 键，滚动鼠标中间滚轮，可以放大的编辑器的字体。



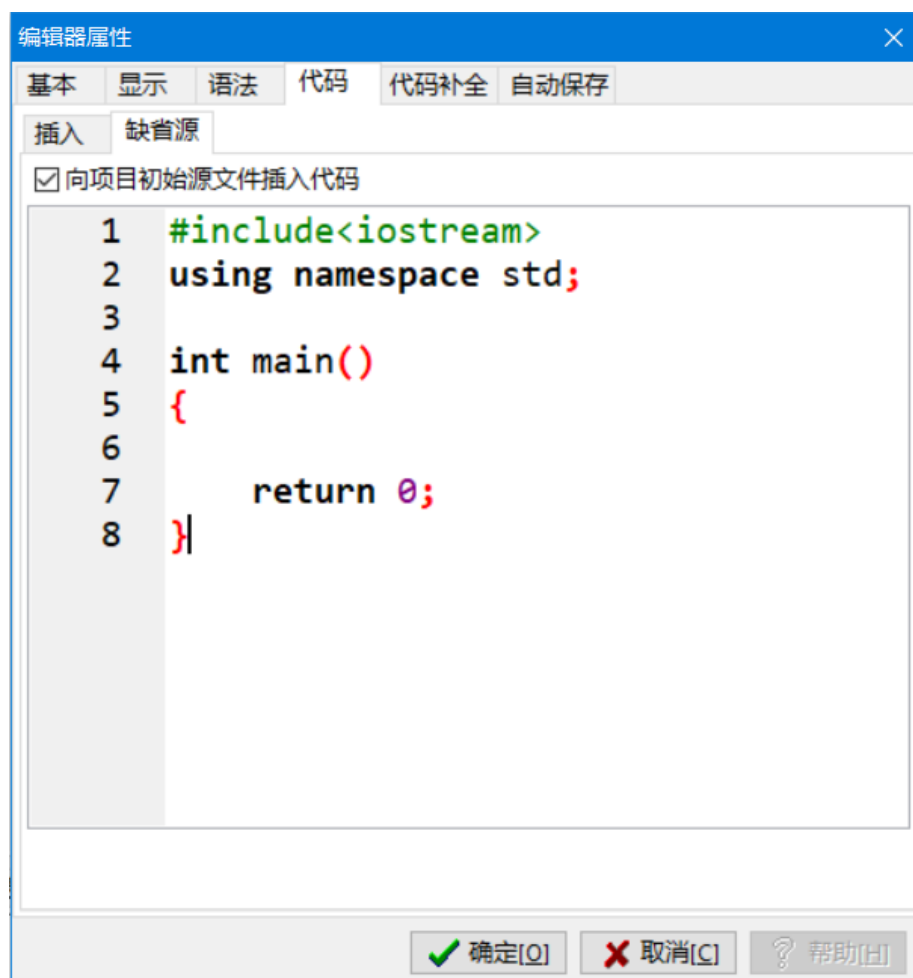
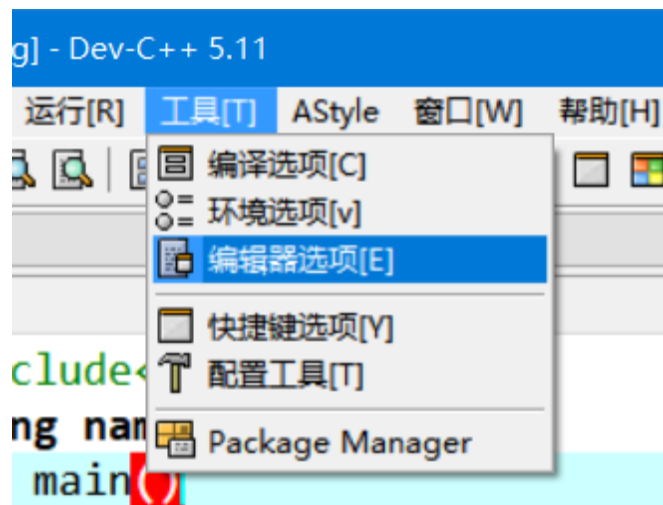
- 方法二：在“工具”菜单里，选择“编辑器选项”菜单项。在编辑器属性里面，调整字体大小。



【添加缺省源代码】



- 步骤一：在“工具”菜单里，选择“编辑器选项”菜单项。
- 步骤二：在“代码”标签页里，输入默认的框架代码。

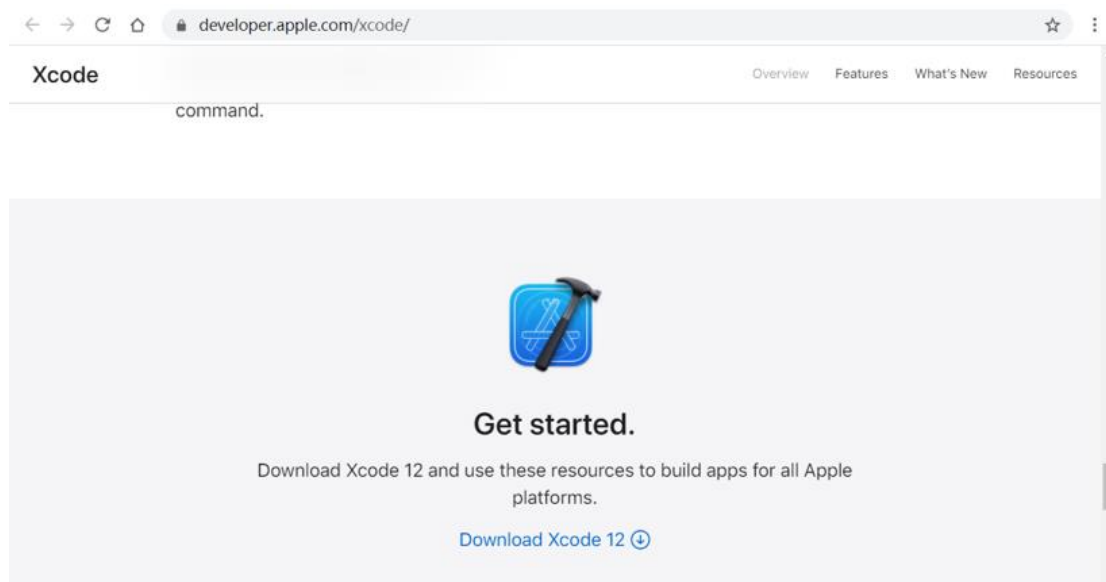




C++开发工具的使用（XCode）

【下载安装】

- 方法一：在 Apple App store 中下载 Xcode。
- 方法二：用浏览器里访问 developer.apple.com/xcode/，点击“Download Xcode 12”链接。



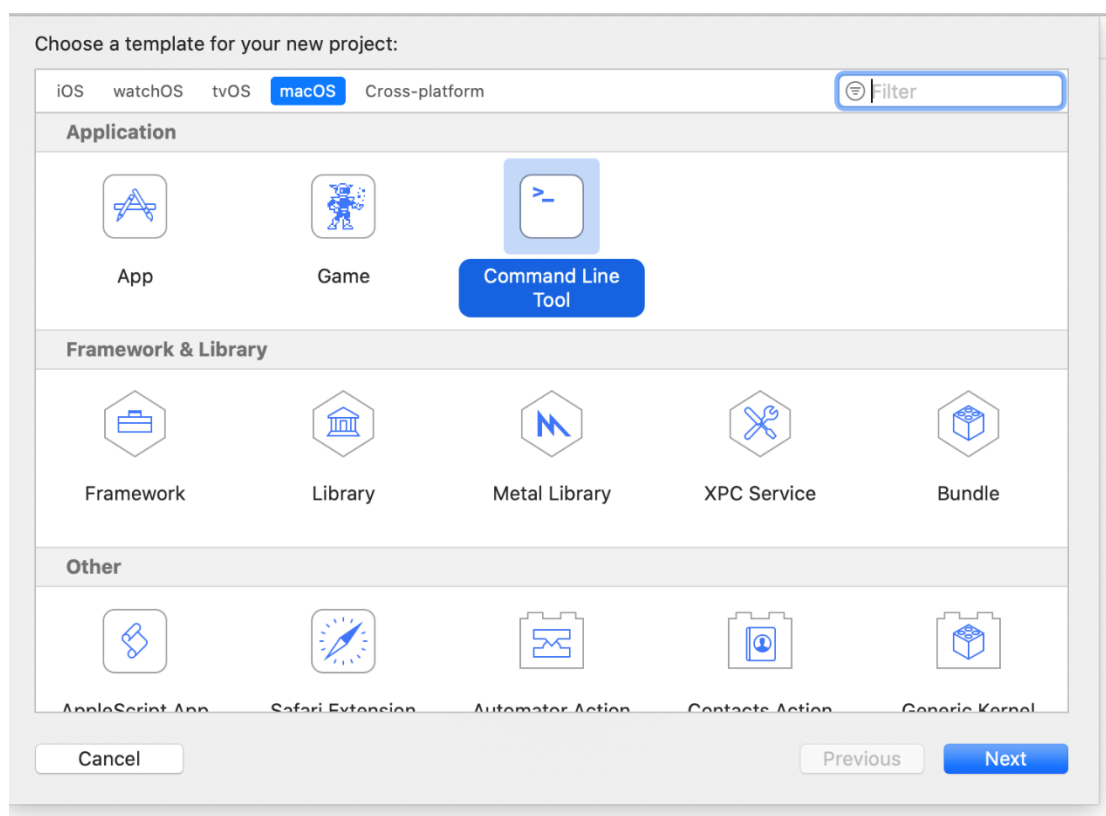
【新建 Xcode 项目】

- 方法一：在 Xcode 启动窗口中选择“Create a new Xcode project”。
- 方法二：在 Xcode 菜单栏中选取“File”→“New”→“Project”菜单项。



【选择项目类别】

- 在“macOS”标签页下，在“Application”分类的列表中，选择“Command Line Tool”。





【设置项目属性】

- 设置项目名称等信息，选择语言为 C++。

Choose options for your new project:

Product Name: CPPDemo

Team: Add account...

Organization Name: Home

Organization Identifier: MyfirstApp

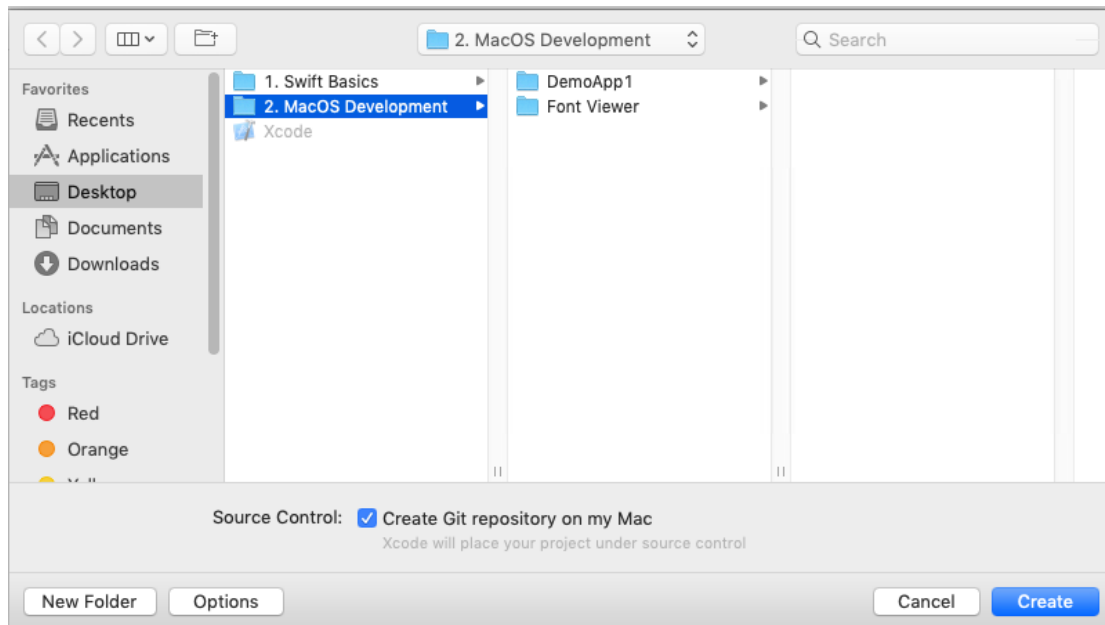
Bundle Identifier:

Language: C++

Buttons: Cancel, Previous, Next

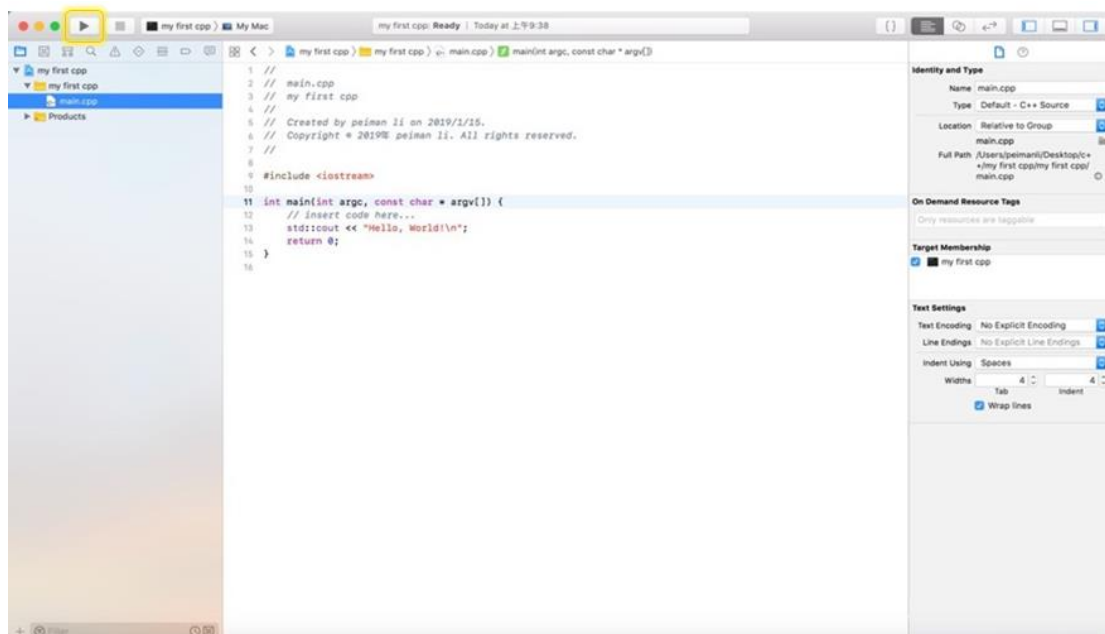
【选择项目保存目录】

- 选择一个目录来保存项目文件，点击“Create”按钮就可以创建项目。



【运行程序】

- 在代码编辑器里编辑好代码。
- 点击工具栏中的三角形图标，即可运行程序。

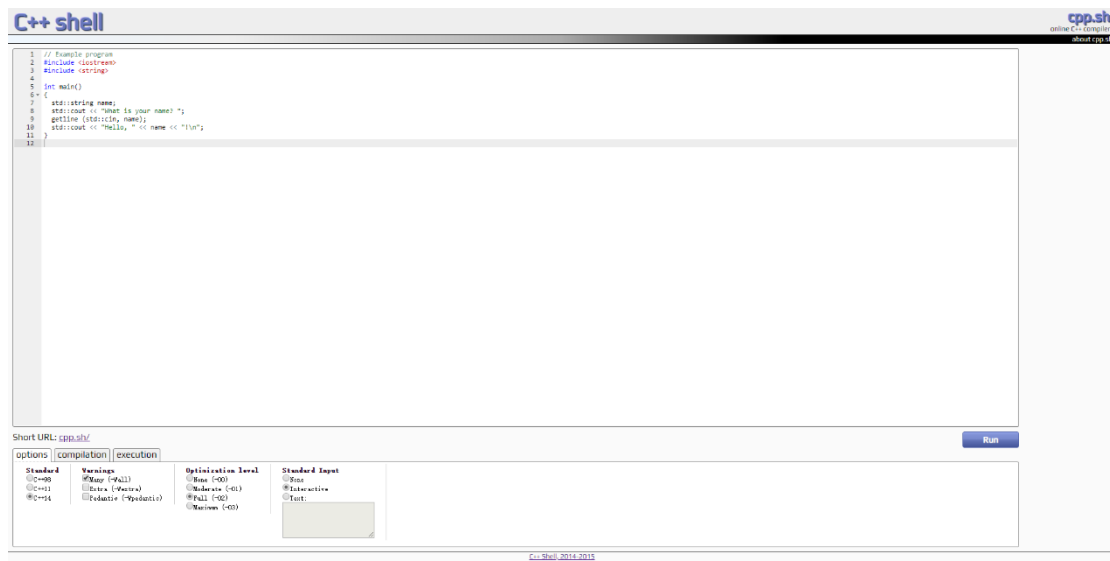




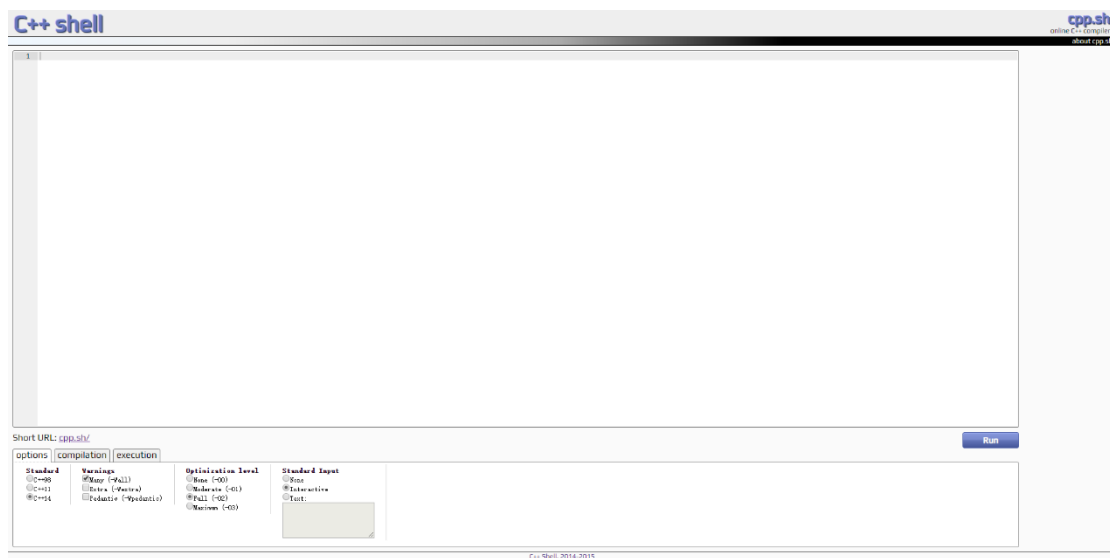
C++开发工具的使用（在线网站）

【在线开发工具】

- 打开浏览器，访问网站地址：cpp.sh，可以在线编辑 C++ 源代码并运行。



【清空当前代码】



【编译错误】



C++ shell

cpp.sh
online C++ compiler
about cpp.sh

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     cin >> a;
6     cout << a;
7     return 0;
8 }
```

Get URL

Run

options | compilation | execution

In function 'int main()':
9:12: error: 'a' was not declared in this scope

C++ Shell 2014-2015

【交互输入】

C++ shell

cpp.sh
online C++ compiler
about cpp.sh

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     char a;
6     cin >> a;
7     cout << a;
8     return 0;
9 }
```

Get URL

Stop

options | compilation | execution

Program running

C++ Shell 2014-2015

【运行结果】



C++ shell

cpp.sh
online C++ compiler
about cpp.sh

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     char a;
6     cin >> a;
7     cout << " " << a << endl;
8     cout << " " << a << " " << a << endl;
9     cout << a << " " << a << " " << a << endl;
10    return 0;
11 }
```

Get URL

Run

options | compilation | execution

####

Build info: 0 (failed program termination)

C++ Shell 2014-2015



第一个 C++ 程序

这是我们的第一个 C++ 程序。

【问题描述】

在屏幕上输出“Hello, Xiaozhi!”。

【参考程序】

```
#include<iostream>
using namespace std;
int main()
{
    cout << "Hello, Xiaozhi!";
    return 0;
}
```

【运行结果】

```
Hello, Xiaozhi!
```

【程序解析】

```
#include<iostream>
```

告诉编译器，本程序需要导入输入输出流标准库<iostream>。

- <iostream>中的 i 代表的是 input（输入）；o 代表的是 output（输出）；stream 是“流”的意思，就是从输入输出设备上读出或写入的字符序列，“流”形象表示出来字符是随着时间顺序生成或消耗的。



- `<iostream>` 包括以下内容的声明和实现: `cout` 是标准输出流对象, 表示向标准输出 (屏幕) 写入数据; `<<` 是输出流运算符。
- 使用上面内容, 必须包括 `<iostream>`, 否则编译出错。

```
using namespace std;
```

- `namespace` 指标识符的可见范围。
- 形象的说就是程序中使用到的标识符是由谁提供的。
- 使用命名空间可以解决属于不同命名空间的相同名字标识符的冲突问题。
- C++ 标准程序库中的所有标识符都属于命名空间 `std`。

```
int main()
```

这里是主函数 (main function) 的声明。

- 每个 C++ 程序都必须有且只有一个主函数。
- 主函数中的内容, 由一对花括号 `{}` 括起来。`{}` 及其中间的内容叫做语句块。每一个以分号结束的行, 叫做一个语句。事实上, 分行写代码只是为了方便人阅读。缩进 (tab) 能使程序更加清晰易读。
- 主函数是所有 C++ 程序的运行的起始点。不管它是在代码的开头, 结尾还是中间, 此函数中的代码总是在程序开始运行时第一个被执行。



- main 后面的一对圆括号(), 表示它是一个函数。C++中所有函数都跟有一对圆括号(), 括号中可以有一些输入参数。圆括号中即使什么都没有也不能省略。
- int 表示主函数执行完毕后, 会返回一个整数。

```
cout << "Hello, Xiaozhi!";
```

cout 表示输出, 告诉计算机把引号之间的字符串送到标准的输出设备 (屏幕) 上。

- cout 是 console out 的缩写。
- cout 在<iostream>库, std 命名空间中声明和实现。

```
return 0;
```

主函数的返回语句。

- 主函数默认用数值 0 表示程序顺利结束, 其他数表示有异常发生。
- 返回语句一般是函数的最后一条可执行语句。同一分支 return 后面的语句不会被执行。



C++的注释

C++的注释有很多种写法。下面展示了不同的方式。我们可以在实际编程过程中选择最合适的方式。

```
// 单行注释的第一种写法
/* 单行注释的第二种写法 */
/*
    程序的注释
    也可以跨行
*/
#include<iostream>
using namespace std;
int main()
{
    cout << "Hello, Xiaozhi!"; //行内注释的第一种写法
    return 0; /* 行内注释的第二种写法 */
}
```

C++注释以 /*开始，以*/终止。也能以//开始，直到行末为止。

C++支持单行注释和多行注释。注释行可以添加在程序的任何位置。

注释的内容是为了方便人对程序的理解，注释中的所有字符会被 C++ 编译器自动忽略。