

Dev-C++安装指北

佳木斯大学ACM实验室 2018级王庆龙

Dev-C++安装指北

Dev-C++简介

安装配置

使用方法

常用热键

其他姿势

使用AStyle自动整理代码

断点调试

Dev-C++简介

Dev-C++是一个Windows环境下的轻量级IDE（集成开发环境），集成了TDM-GCC编译器、GDB调试器以及AStyle格式整理器等功能。现由Orwell公司开发，使用GPLv3开源协议。

最新版本：Dev-C++ 5.11 TDM-GCC 4.9.2

最后更新时间：2015-04-27

安装配置

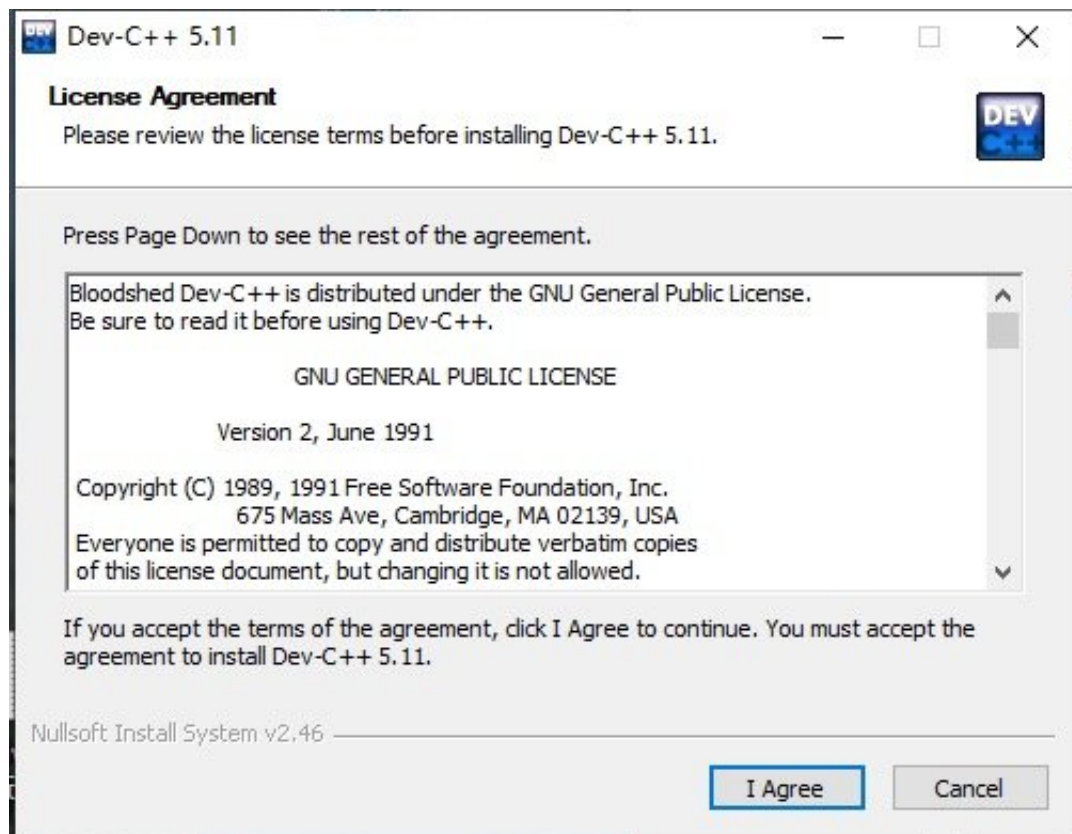
1. 下载并打开2019ACM新生集训群群文件中的 **Dev-C++ 5.11 TDM-GCC 4.9.2 Setup.exe**

文稿	AirDroid_Desktop_Client_3.6.5.0.dmg	2019年7月13日 下午9:09
下载	Alfred_4.0.3_1092.dmg	2019年7月16日 上午9:18
影片	aria2.conf	2019年9月1日 上午11:29
音乐	dell_p2414h.icc	2019年9月5日 下午9:57
图片	Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe	今天 下午8:51
Video	iTerm.app	2019年4月20日 上午11:04
Cloud	ShadowsocksX-NG-R8.dmg	2019年6月25日 下午1:46
iCloud 云盘	TheUninstaller.dmg	2017年7月28日 下午3:27

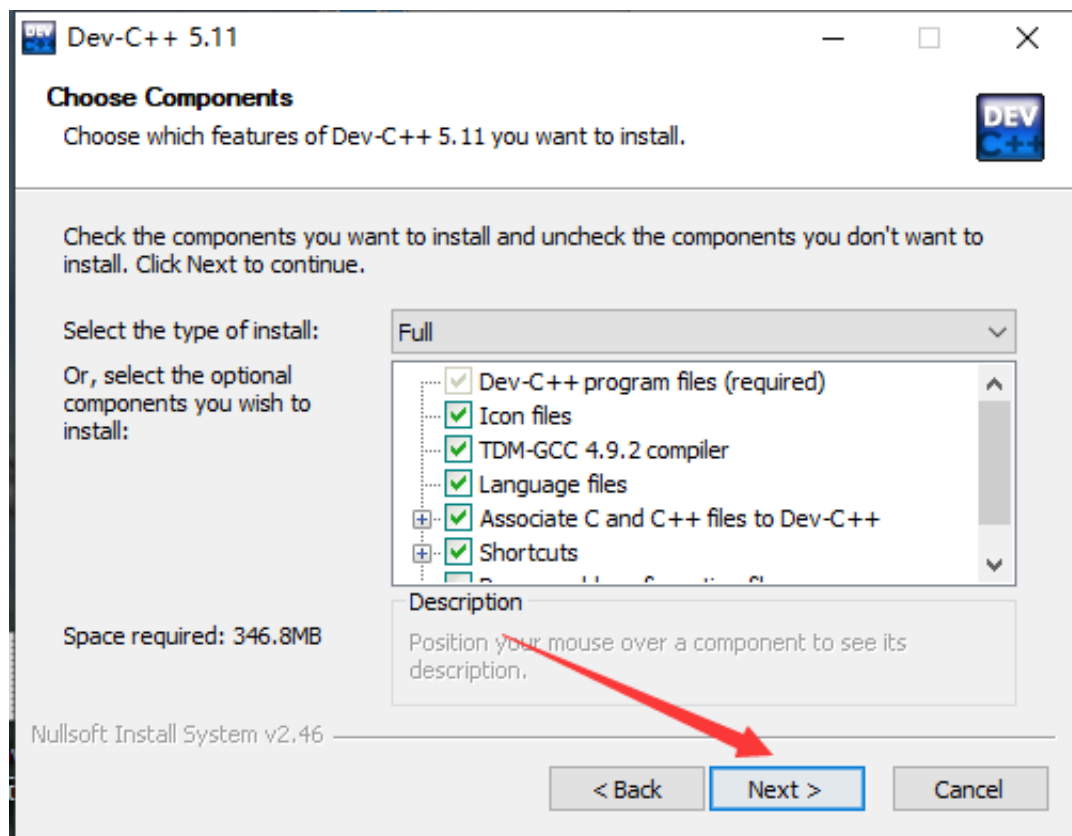
2. 程序将弹出语言选择窗口，点击OK即可



3. 提示阅读软件许可协议，点击 **I Agree** （我同意）



4. 提示选择要安装的组件，请选择 **Full** （完全安装），点击 **Next** （下一步）

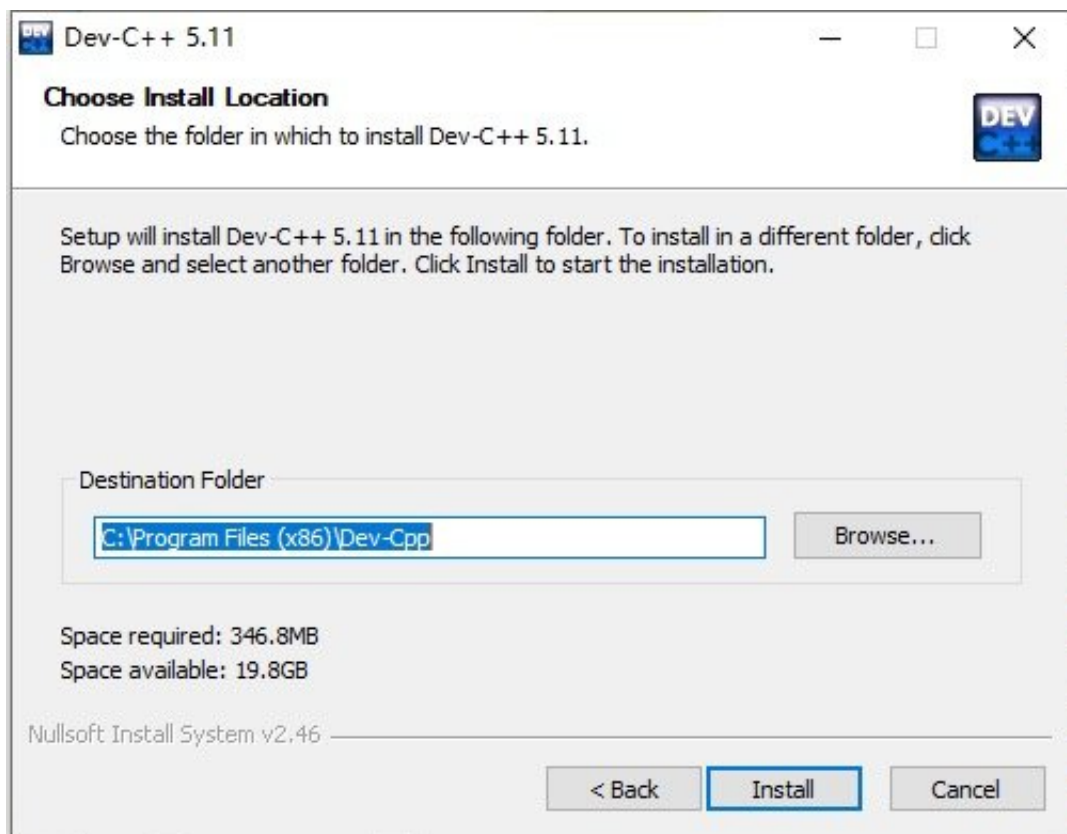


5. 选择安装路径，如不想安装到C盘，可以在路径框中直接修改安装路径或点击 **Browse**（浏览）选择想要安装到的文件夹。

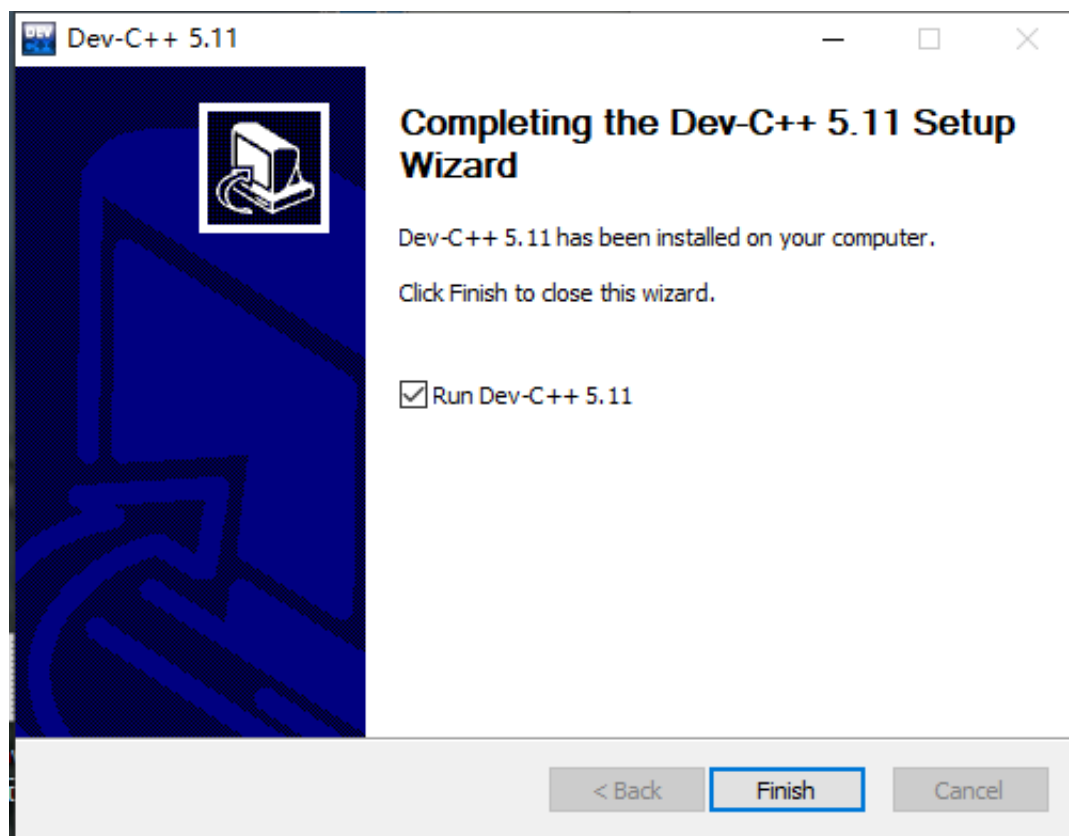
如，可更改路径为： **D:\Program Files (x86)\Dev-Cpp** 即可安装到D盘默认文件夹下；

或更改为： **D:\MySoftware\Dev-Cpp**，即可安装到D盘的MySoftware文件夹下

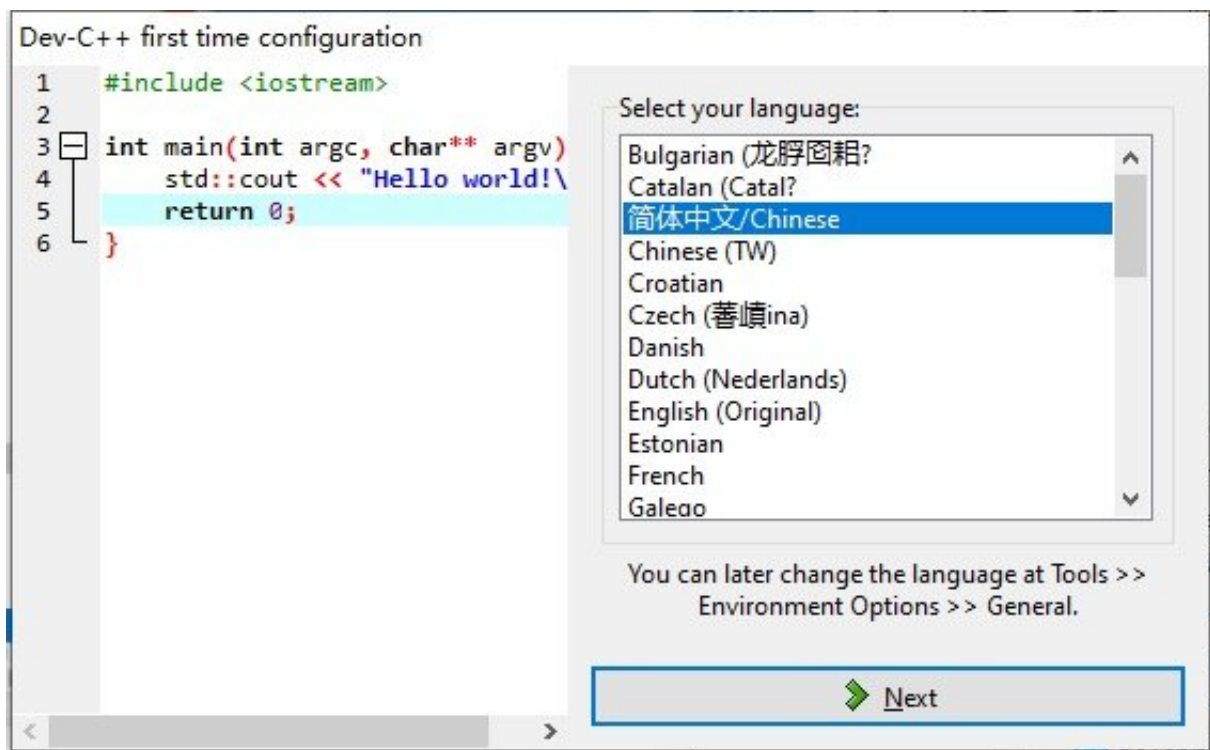
安装路径请勿使用名称中带有中文的文件夹



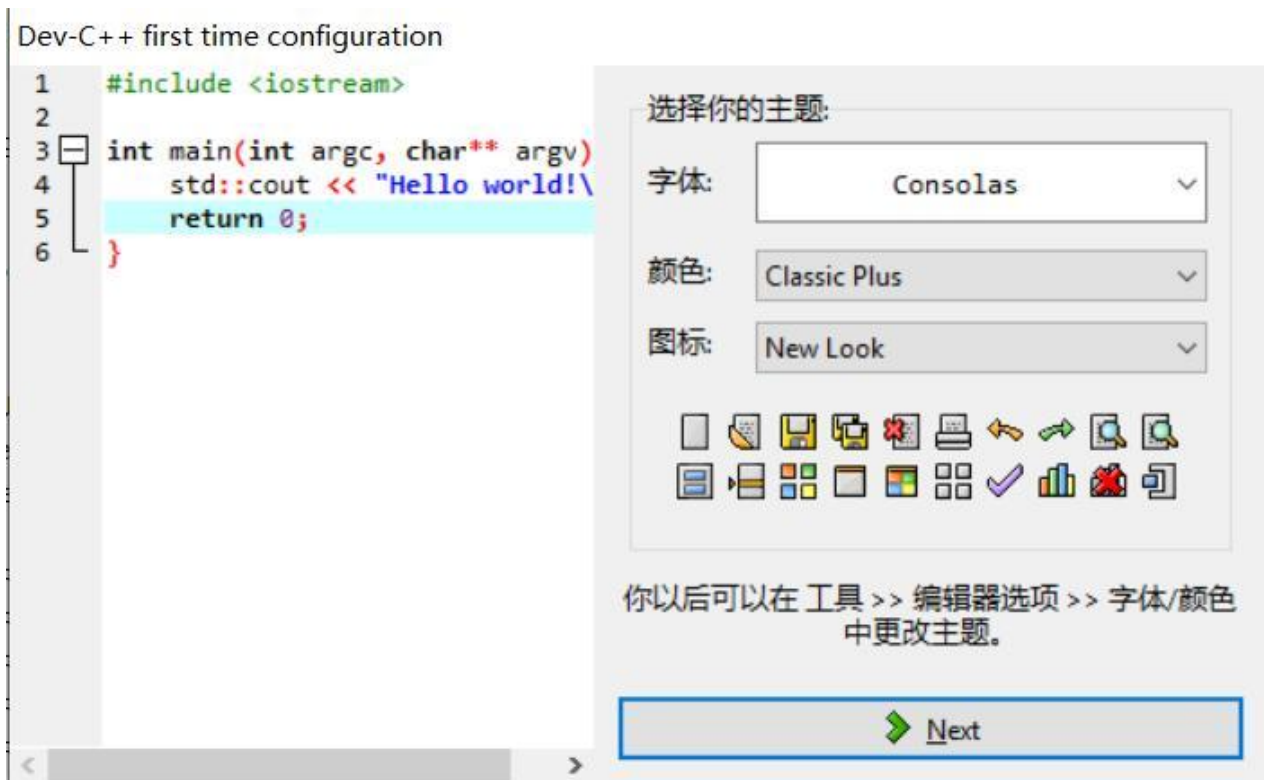
6. 等待程序安装结束后，将提示Dev-C++安装完成，点击 **Finish**（完成）可结束安装，Dev-C++将自动运行（如取消勾选 **Run Dev-C++ 5.11** 结束后将不会自动运行）。至此，安装过程结束。



7. 首次运行，程序将提示选择显示语言，在右面的列表中选择 **简体中文/Chinese**

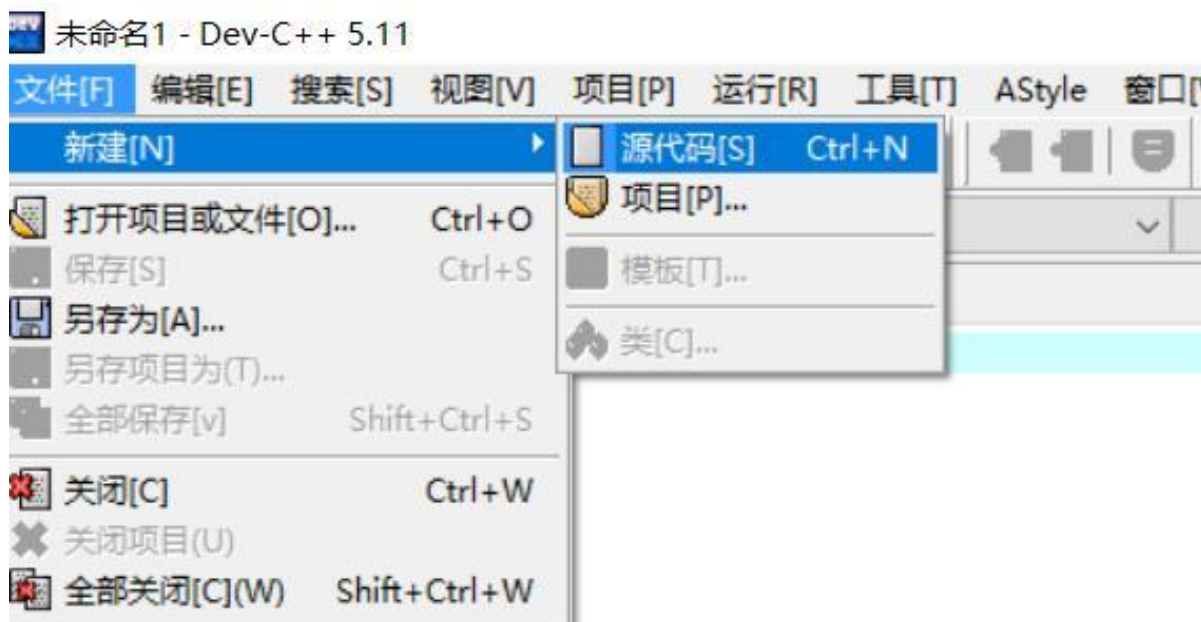


8. 之后会提示更改外观配置，可以选择自己喜欢的字体、代码高亮配色以及图标方案。选择完成后点击 **Next**，再点击 **OK** 即可。

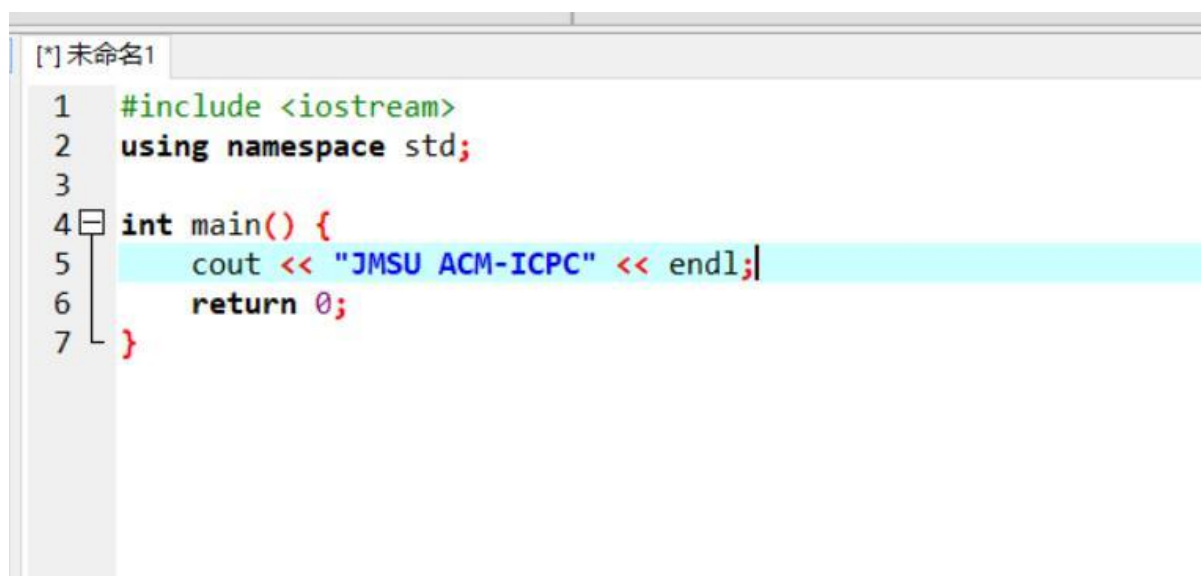


使用方法

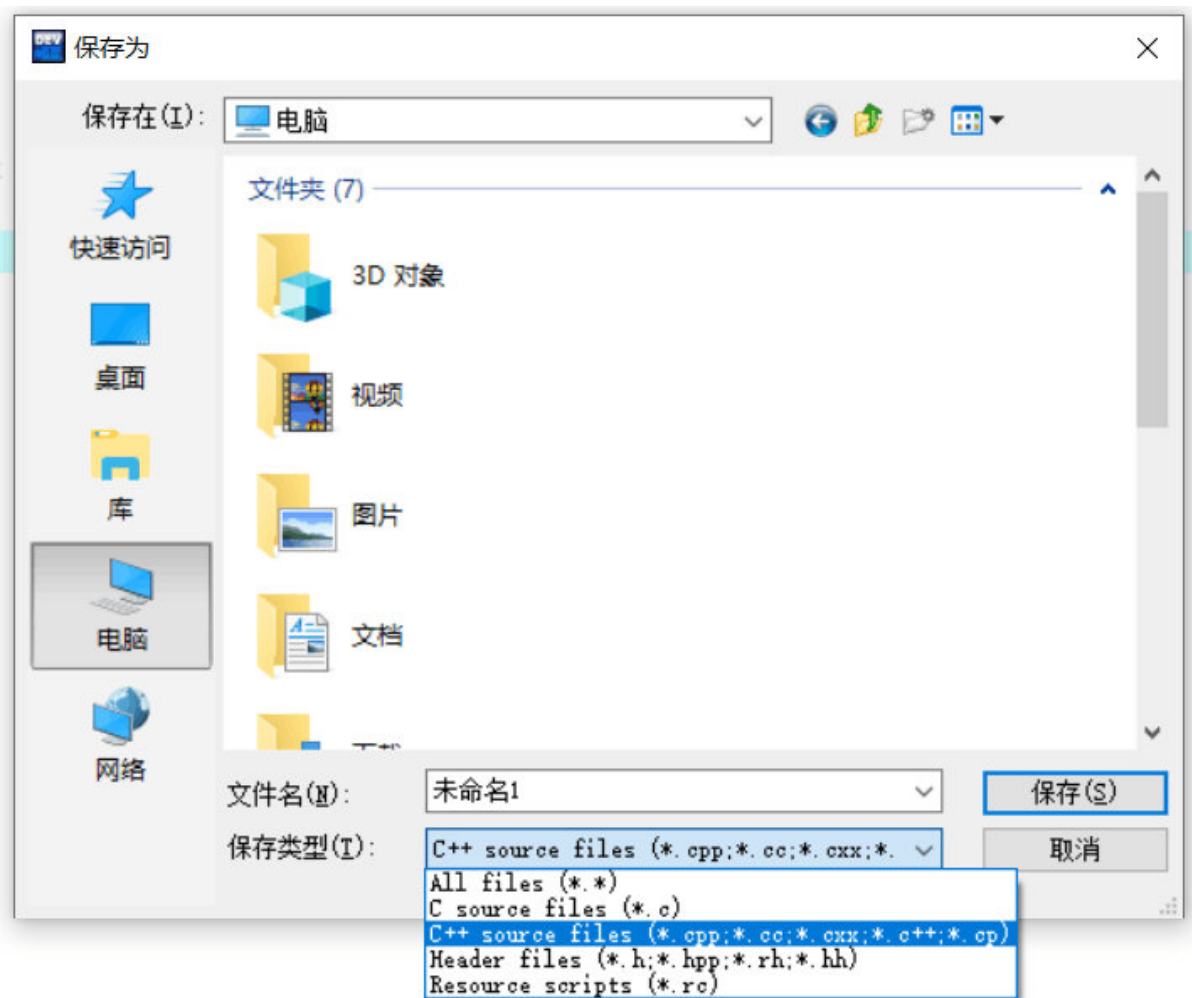
1. 点击文件 > 新建 > 源代码或按下Ctrl + N即可新建一个文件



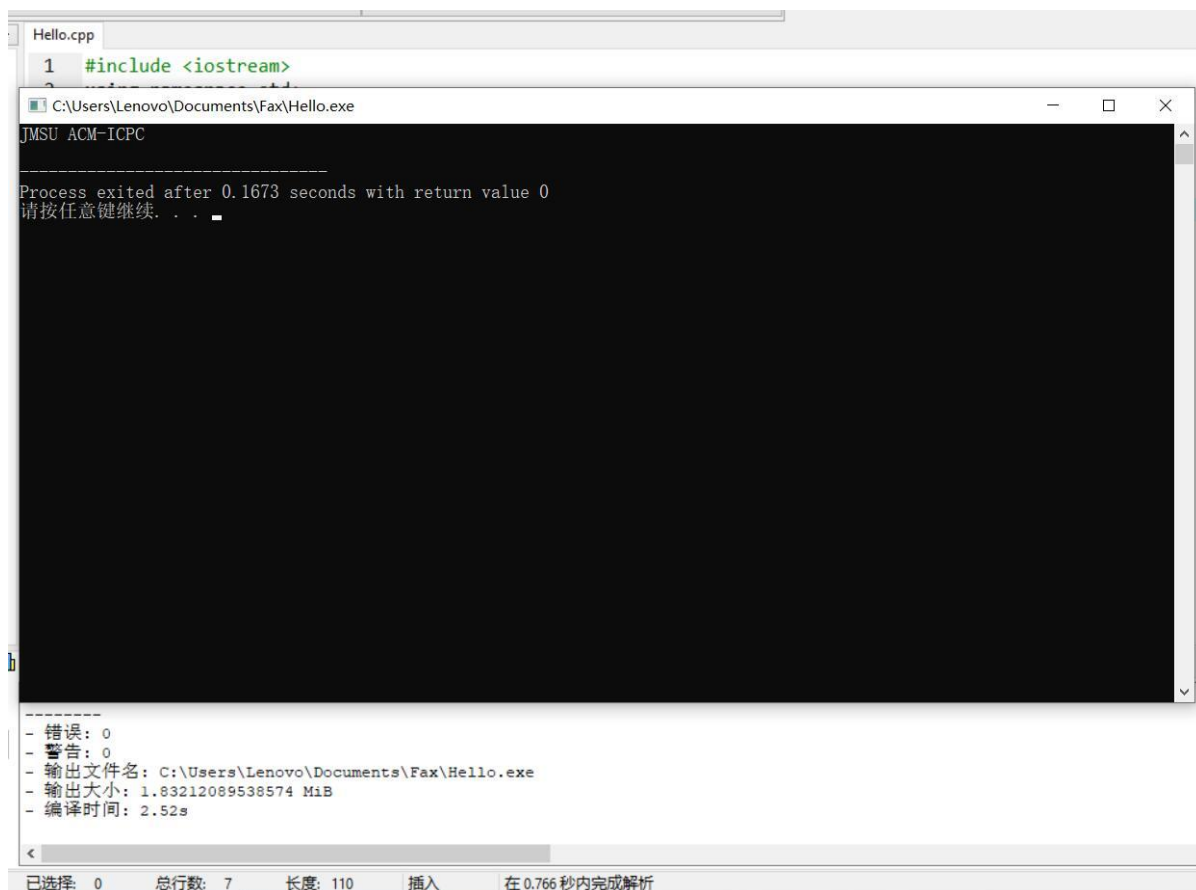
2. 之后正常编写你的代码文件，可以按住Ctrl键并滚动鼠标滚轮来调整编辑器字体大小。
文件名旁的星号，代表本文件未保存，可以点击保存图标或使用Ctrl + S保存文件



3. 初次保存未保存的文件，会询问你保存的位置以及文件类型
C代表C语言源代码， C++代表C++源代码， Header代表.h头文件， Resource为资源文件



4. 保存完成后，点击编译运行图标或按F11编译并运行，当编译进程结束后，将弹出运行结果的命令行窗口，可在此窗口内进行输入以及观察运行结果，全部执行完后按任意键退出。



常用热键

Tab 缩进

F9 编译

F10 运行

F11 编译并运行

Ctrl + **Home** / **End** 光标跳转到当前文本的开头/末尾处

Ctrl + **D** 删除光标所在行的整行文本

Ctrl + **/** 使用 `//` 注释掉光标当前所在行

Ctrl + **Shift** + **A** 使用AStyle自动格式化代码

Ctrl + **S** 保存文件

Ctrl + **N** 新建文件

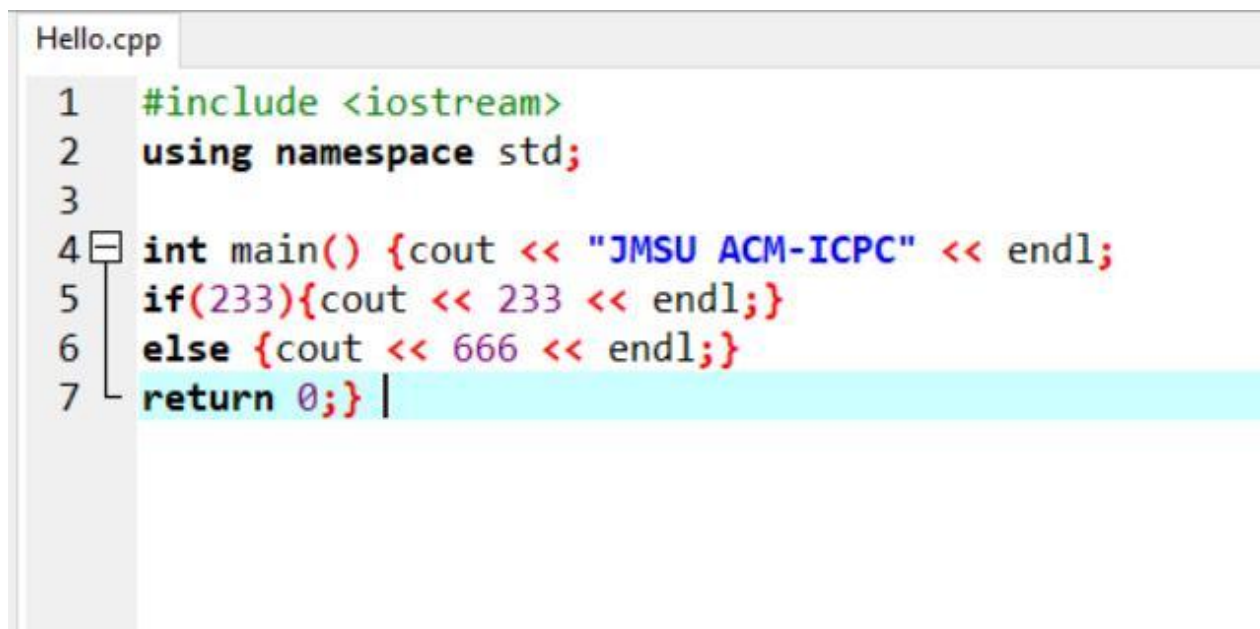
其他姿势

使用AStyle自动整理代码

使用Dev-C++内置的AStyle，可以按照预定的代码风格/缩进格式自动整理你的代码排版。

再强大的工具，也不如养成习惯。建议预先学习代码格式规范

1. 如果你的代码没有缩进，没有准确地换行



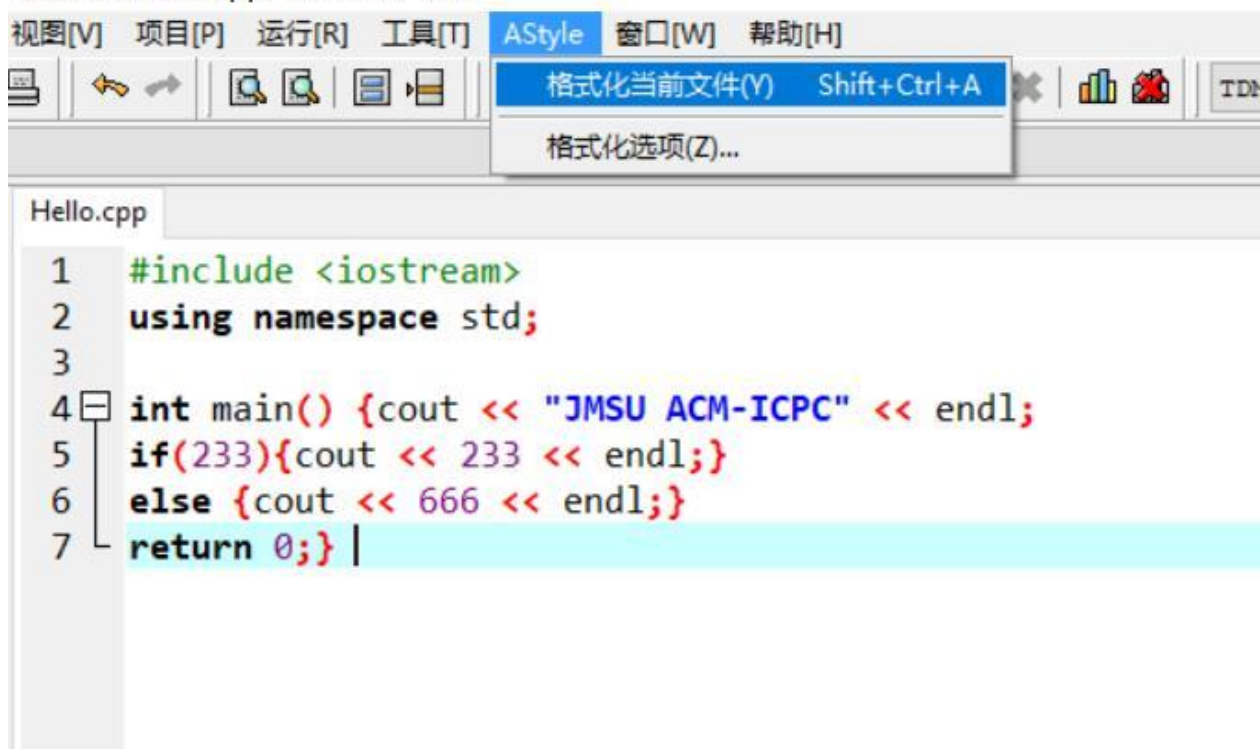
The screenshot shows a code editor window titled 'Hello.cpp'. The code is as follows:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {cout << "JMSU ACM-ICPC" << endl;
5  if(233){cout << 233 << endl;}
6  else {cout << 666 << endl;}
7  return 0;} |
```

The code is not formatted with proper indentation or line wrapping. The last line is highlighted in light blue.

2. 按下Ctrl + Shift + A或点击菜单栏中的AStyle > 格式化当前文件来整理代码

ients\Fax\Hello.cpp - Dev-C++ 5.11



The screenshot shows the Dev-C++ interface. The 'AStyle' menu is open, showing the option '格式化当前文件(Y) Shift+Ctrl+A'. The code in the editor is now formatted as follows:

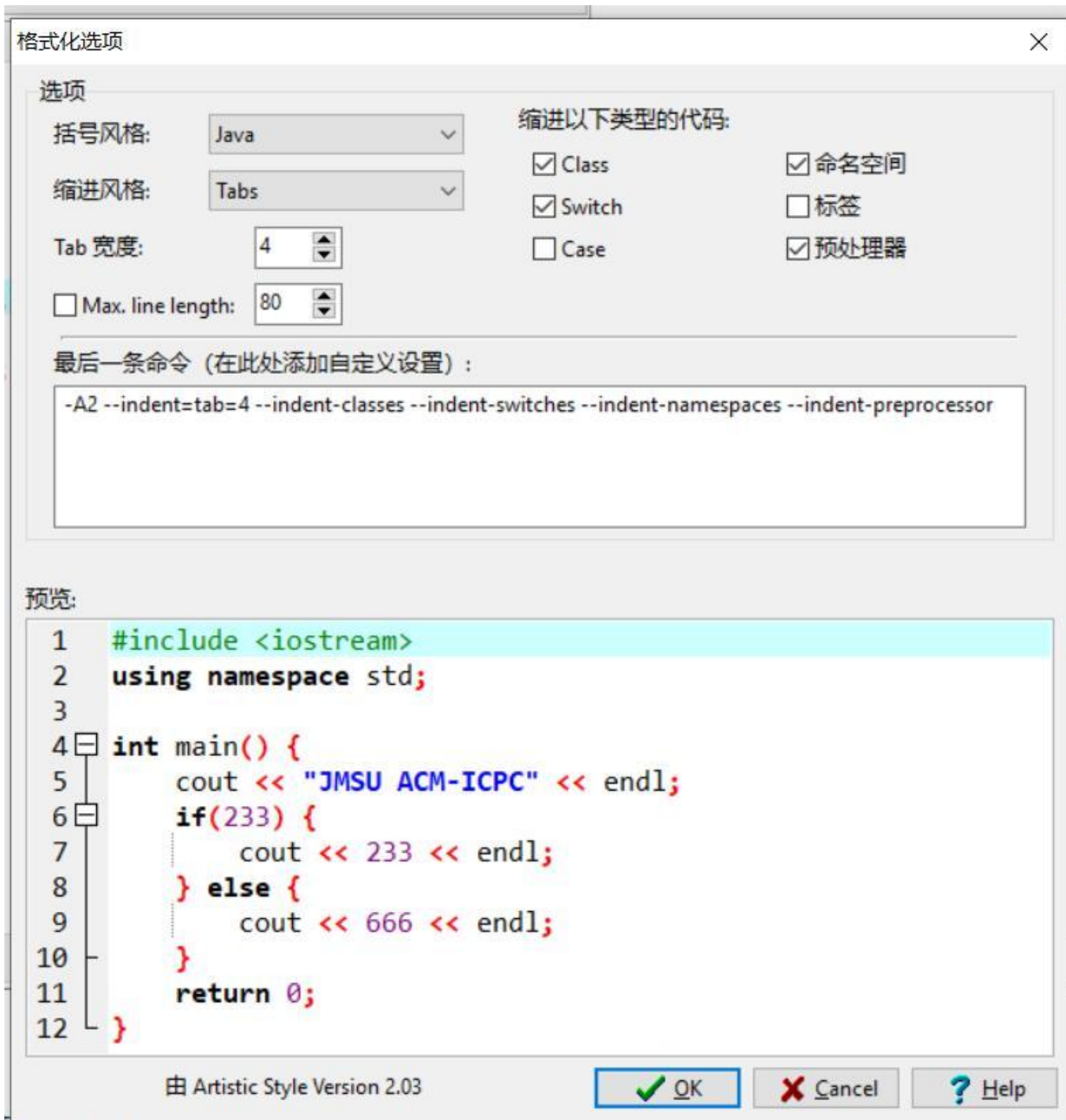
```
1  #include <iostream>
2  using namespace std;
3
4  int main() {cout << "JMSU ACM-ICPC" << endl;
5  if(233){cout << 233 << endl;}
6  else {cout << 666 << endl;}
7  return 0;} |
```

The code is now properly formatted with consistent indentation and line wrapping. The last line is highlighted in light blue.

3. 代码将按照预定的设置自动整理你的代码

```
[*] Hello.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      cout << "JMSU ACM-ICPC" << endl;
6      if(233) {
7          cout << 233 << endl;
8      } else {
9          cout << 666 << endl;
10     }
11     return 0;
12 }
```

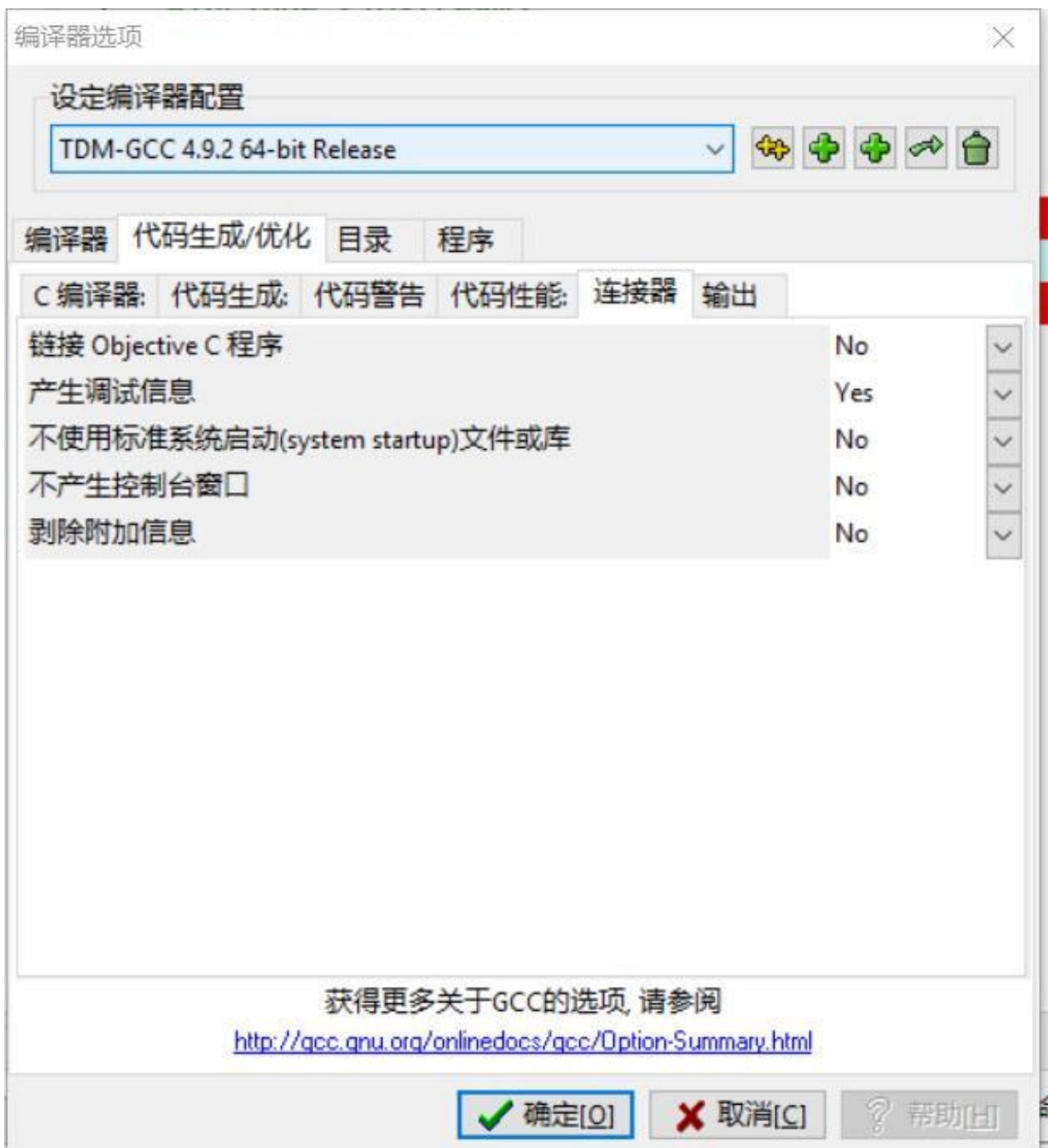
4. 你可以在Astyle > 格式化选项中调整代码风格



断点调试

使用Dev-C++内置的GDB调试器，来断点调试你的代码并监视变量变化。

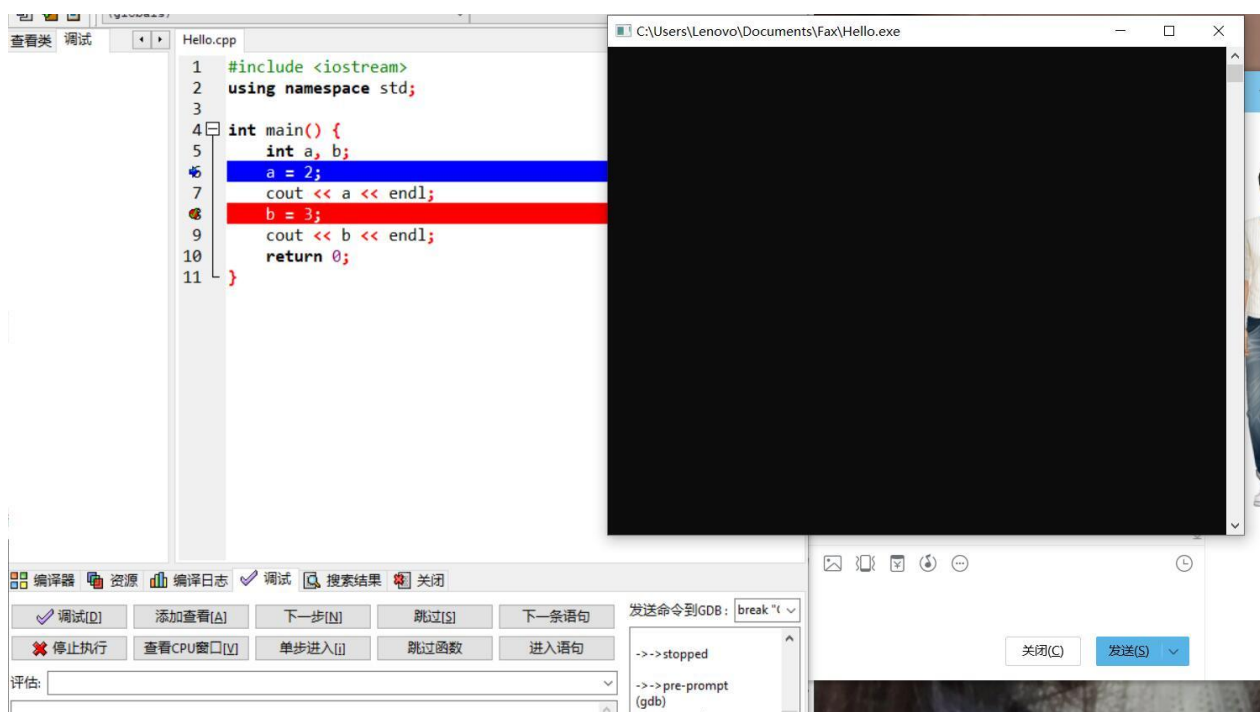
1. 首先配置调试信息，点击工具 > 编译选项，点击代码生成/优化，点击连接器，将 **产生调试信息** 修改为Yes，点击确定使设置生效。



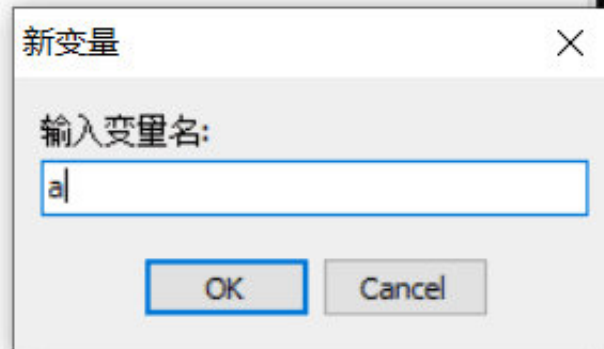
2. 设置后编译的cpp文件会同时生成调试信息便于断点调试，在代码中你想要设置断点的地方点击左侧行数字，该行变为红色，代表调试时程序将在此处暂停执行等待调试操作。

```
Hello.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a, b;
6      a = 2;
7      cout << a << endl;
8      b = 3;
9      cout << b << endl;
10     return 0;
11 }
```

3. 点击下方的调试选项卡，先将程序编译后点击调试按钮，将弹出命令行窗口。代码行变为蓝色代表程序执行到此处并已经在断点处暂停了运行。



4. 点击添加变量按钮，添加你想要监视的变量



5. 点击下一步，即可运行当前行代码跳至下一行，可以在左侧的调试窗口看到变量的数值变化。

