

第一部分 初识C++——顺序结构

准备课 轻、便、快的C++学习神器（运用Dev-C++）

第1课 被玩坏的字符（程序的输出）

第2课 蹩脚的“ChatGPT”（信息输入）

第3课 舞动的机器人（趣味输入应用）

第4课 一桩大买卖（运算符、变量）

第5课 这面积总缺那么一点（浮点数数据类型）

第6课 睡不着就数羊（计数）

第7课 星号金字塔（双变量累加）

第8课 可怕的核废水（半衰与阶乘）

第9课 陈醋和酱油不能混为一瓶（变量值的互换）

第10课 神秘的摩斯密码（system()和Beep()函数）

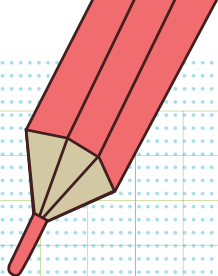
第11课 安排出游车辆（取余数）

第12课 小数字大学问（进制）

第13课 ASCII编码背后的秘密（ASCII编码）

第14课 列竖式做计算（setw()函数）





准备课

轻、便、快的 C++ 学习神器 (运用 Dev-C++)

学习 C++，我们将进入一个全新的领域。与计算机交朋友，在应用数学的海洋里遨游，探寻发明者的智慧结晶。别觉得 C++ 很难，其实你早已是一个编程高手，面对如此复杂的上学指令，你每天都游刃有余地执行着，不是吗？

上学指令：

起床



换衣服



洗漱



吃早餐



其中每个大的指令又包含很多小的指令

整理书包 → 拿出书包 → 拉开拉链 → 把书放进去 → 拉上拉链



出发

你多厉害呀，既能编写命令，又能执行命令。

只是现在的计算机还不够智能，它没办法理解我们说的话。下面的两段对话，就能把计算机绕晕了。

A

我：你等等，我先去方便一下。

计算机：方便是什么意思？

我：方便就是上厕所的意思。

计算机：我懂了！

B

我：我方便的时候，你随时过来。

计算机：不太好吧，你上厕所的时候我过来太不文明了吧！

我：额~这里说的“方便”是我有时间的意思。

计算机：什么情况，我懵圈了。



因此，与计算机沟通需要借助编程语言。那么，如何将编程语言表达给计算机呢？这就需要使**集成开发环境**，在其中写入编程语言，经过**编译**转换成计算机能够看懂的命令，然后运行。

Dev-C++就是这样一个集成开发环境，运用它可以实现C++程序的**编辑**、**编译**、**运行**和**调试**。如果你的计算机中还没有安装这个软件，记得先下载并安装。

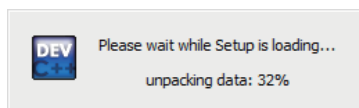
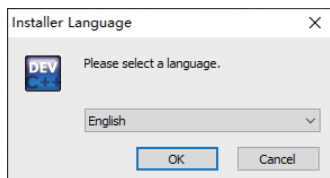
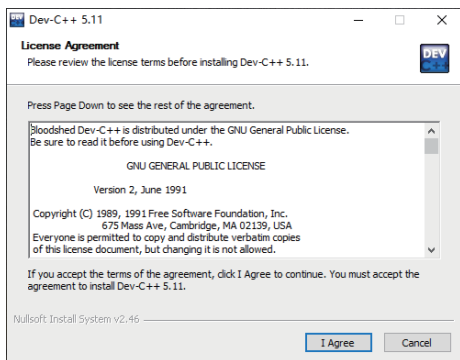
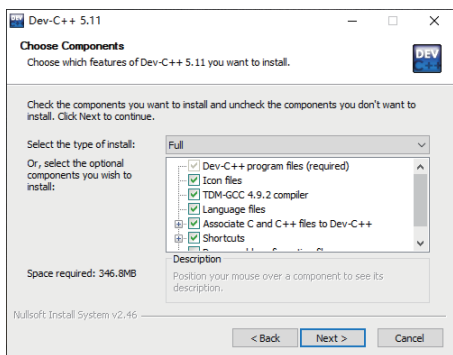
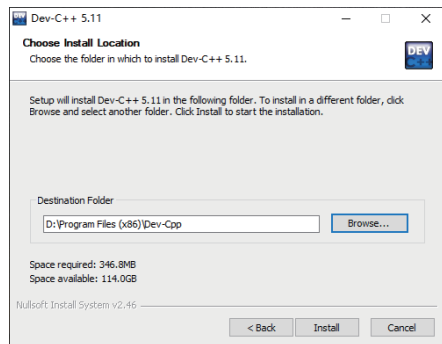
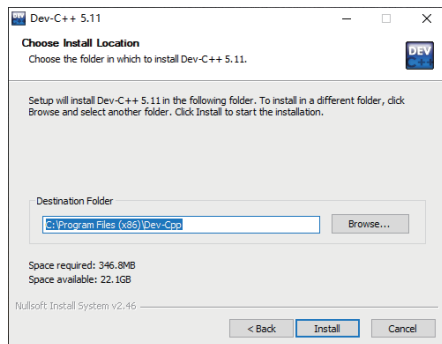
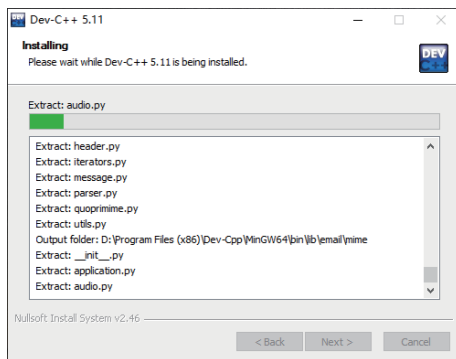
随着时代的进步和ChatGPT的发展，等你学有所成，或许就能让计算机直接理解我们的自然语言。我真期待那一天的到来，由你来创造历史。



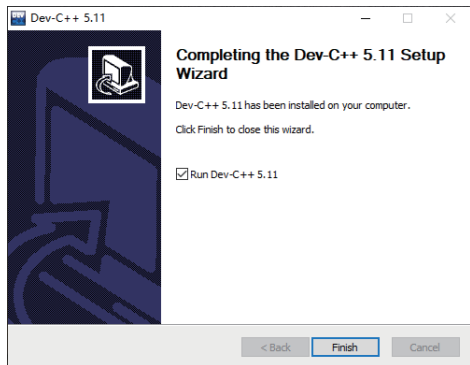
▼ Dev-C++的安装

1 双击运行Dev-C++安装包。

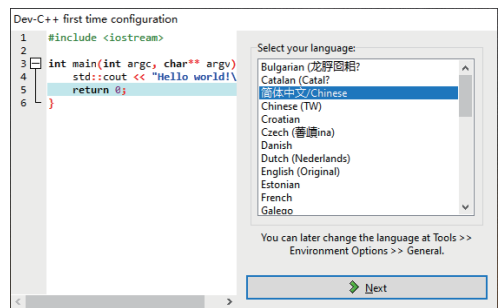
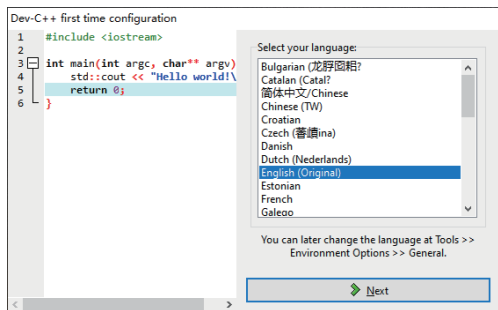


2 等待安装程序加载。**3** 加载完成后，选择安装环境的语言，默认为English。**4** 单击“I Agree”按钮接受许可协议，进入下一步。**5** 单击“Next”按钮，继续下一步。**6** 选择该集成开发环境要安装的磁盘位置，这里我将该软件安装在D盘。**7** 单击“Install”按钮后，等待安装进行。

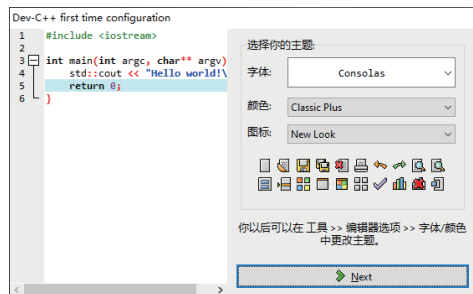
- 8 单击Finish按钮完成安装，并打开Dev-C++。



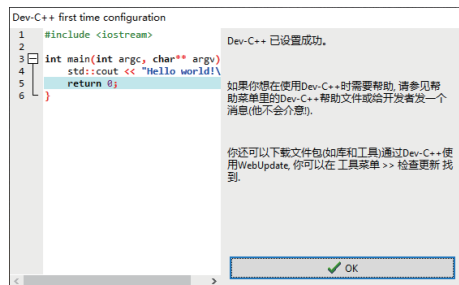
- 9 进入软件界面后选择语言，这里我选择的是“简体中文/Chinese”，然后单击Next按钮。



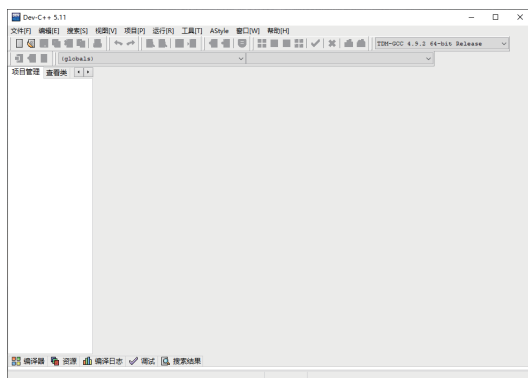
- 10 这里可以调整字体和风格，通常习惯使用默认状态，直接单击Next按钮，进入下一步。



- 11 一切完成，单击OK按钮开始编程。



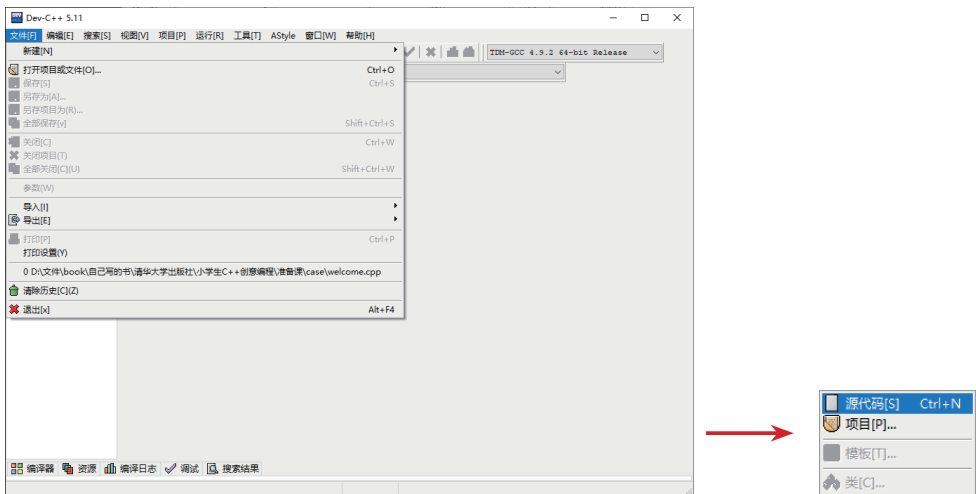
- 12 进入编程界面，它长这样。



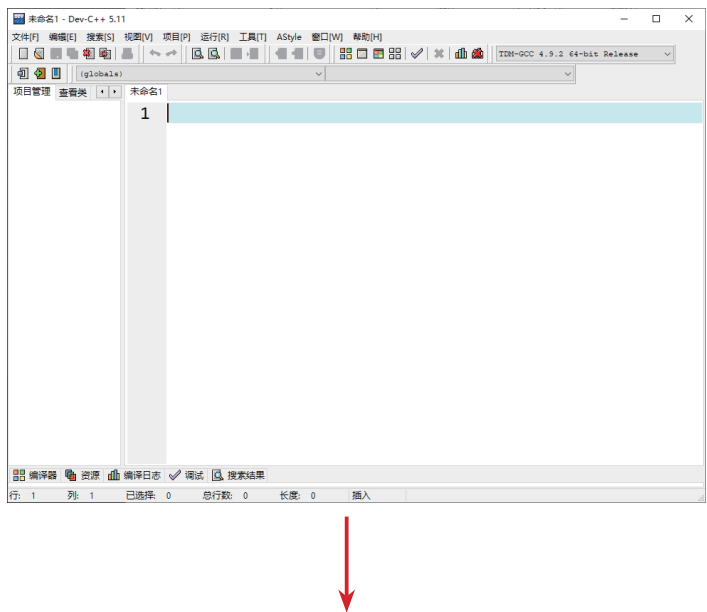
▼ 我的欢迎程序

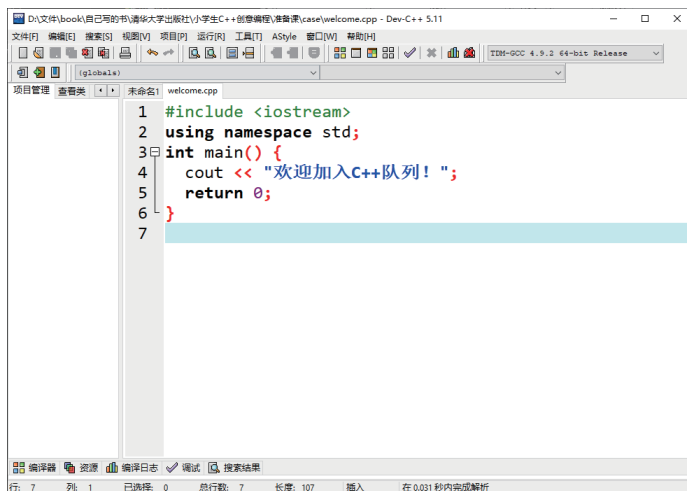
尝试着创建C++源文件，编写一个程序。编程学习过程中，能够让程序正常运行至关重要，如果程序卡住了，不要着急往下学习，先解决程序问题。**理解知识**和**实操编程**两者相辅相成，缺一不可。

1 依次单击“文件”→“新建”→“源代码”。



2 在光标闪烁的位置输入代码。





- 3 输入以下代码，这里不用理解代码的含义，照着敲一敲，感受一番，先成功运行第一个C++程序。

注意

代码中的符号，要在英文格式下输入！

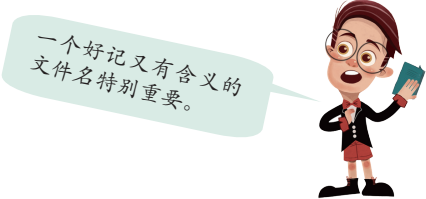
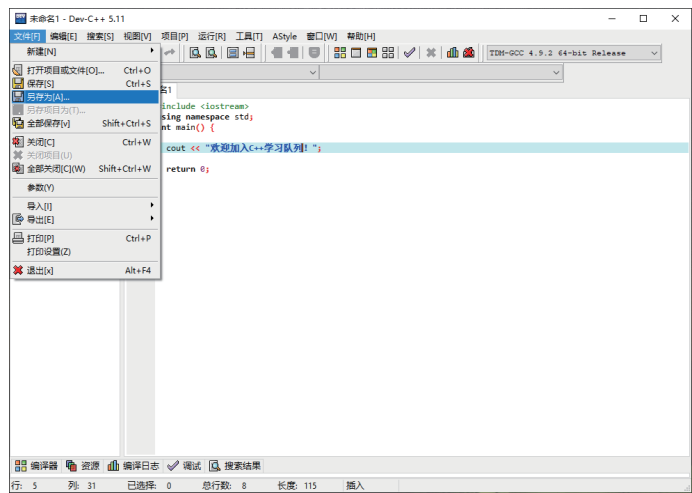
代码

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     cout << "欢迎加入C++队列! ";
5     return 0;
6 }
```

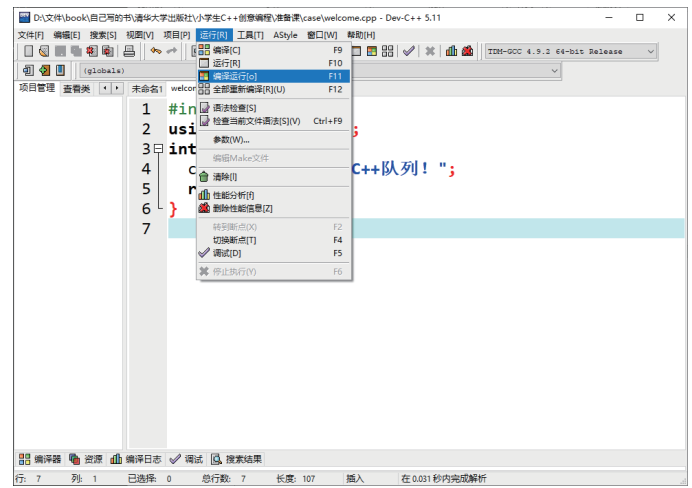


第一个程序也不一定要
“Hello World!” 嘛！

- 4 运行程序之前必须保存程序文件（也称为源代码文件或源程序文件），单击“文件”→选择“保存”或“另存为”，将程序文件命名为welcome.cpp并保存到计算机中。



5 源代码文件保存后，依次单击“运行”→“编译运行”，对源代码进行编译运行。



此时一共进行了两步，一步是编译，一步是运行。我们也可以将它拆开，先编译再运行。



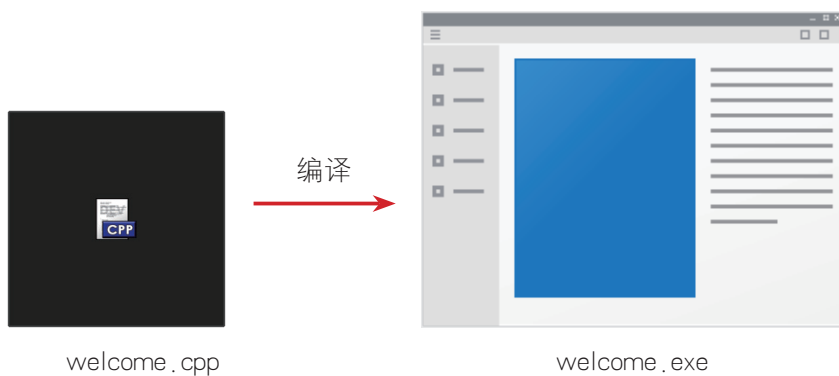
敲黑板

编写完代码后，单击**编译运行**，查看程序结果。

源文件 → 编译 → 运行



源程序文件编译后会生成一个 **.exe** 文件，运行的就是这个 **.exe** 文件。



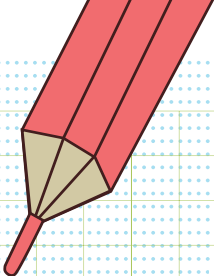
注意

每次修改代码后，运行程序前都需要编译。

6 程序运行效果。



试试双击 `welcome.exe` 文件，看看程序是否直接运行。



第1课



被玩坏的字符（程序的输出）

大脑是一个超级强大的计算机，我们的阅读、游历、见识是输入，而我们的好奇、思考和想象则是处理，我们的写作、绘画和编程则是输出。

古时候李白远望庐山瀑布（输入），这个信息经过他的大脑进行思考和创作（加工处理），他便能赋诗一首（输出）。

《望庐山瀑布》

日照香炉生紫烟，遥看瀑布挂前川。
飞流直下三千尺，疑是银河落九天。

► 温故知新

将我们准备课编写的代码修改一番，就能输出《望庐山瀑布》。

1 创建一个新的C++源代码文件，命名为`poem.cpp`。



敲黑板

在计算机系统中，每个文件都有一个名字，而这个名字的命名有个规则：**文件名** + **++扩展名**。

`poem.cpp`

↑ ↑
文件名 扩展名

就像我们的命名规则是**姓**+**名**一样。

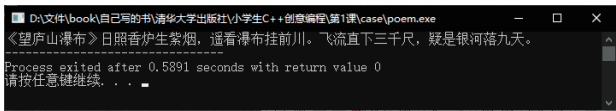
- 文件名：用于识别文件，通常需要起一个既易于记忆又能准确反映文件内容的名称。
- 扩展名：告诉计算机这个文件属于何种类型，比如图片的JPG类型、视频的MP4类型等。

2 好了，让我们开始编写代码吧。

代码

```
1 #include <iostream> //头文件
2 using namespace std; //命名空间
3 int main() { //主函数
4     cout << "《望庐山瀑布》"; //输出语句
5     cout << "日照香炉生紫烟，遥看瀑布挂前川。"; //输出语句
6     cout << "飞流直下三千尺，疑是银河落九天。"; //输出语句
7     return 0; //返回0
8 }
```

3 运行程序，诗句就输出到屏幕中了。



一起来分析一下这个程序，这是我们学习C++的起点。

(1) #include <iostream> 预处理命令。

翻译助力理解

- include：包括、包含。
- iostream：“i”代表输入（input），“o”代表输出（output），“stream”表示数据流。组合起来可以理解成输入输出数据流。

在我们使用C++之前，前辈们已经创造了许多工具，使得我们的编程变得更加便捷。例如，**iostream**就是一个拥有强大的输入和输出能力的工具，其同名文件被称为“**头文件**”。

想要将这个“头文件”对应的工具运用到程序中，就需要通过**#include**这个**预处理指令**将“头文件”包含到我们的程序里，为我所用。





划重点

这句代码的意思就是告诉计算机，我要将 `iostream` 里面的输入和输出功能运用到我的程序中。

语法规则是：`#include <头文件的名字>`。

(2) `using namespace std`。

翻译助力理解

- `using`：使用、运用。
- `namespace`：命名空间。
- `std`：`standard`的缩写，意思是标准。

随着越来越多的人为C++创造工具，各种工具的名字就很有可能重复。为了避免重名带来的冲突，于是就引入了命名空间。

想一想：计算思维用于生活

为什么学校在分班的时候，要把两个名字相同的学生分在不同的班级呢？

在“C++学校”，有两名学生的名字都叫作“凤飞”，如果他们分在同一个班，当老师点名“凤飞”的时候，这两个同学就不知道喊的是谁了。

如果将一名“凤飞”同学分在A班，将另外一名“凤飞”同学分在B班，是不是就解决问题了。即使学校点名时，只需说“A班的凤飞”，也可以轻松地区分。

这里的A班、B班就可以看作命名空间。

使用 `using namespace std` 这句代码就是告诉编译器，“我要在代码中使用标准命名空间中的工具”。这样就可以直接使用工具 `cout`，而不需要在前面添加 `std::`。

如果没有了命名空间，就需要在 `cout` 前加上标识 `std`，如下所示。

代码

```
1  #include <iostream>           //头文件
2  int main() {                 //主函数
3      std::cout << "《望庐山瀑布》"; //输出语句
4      std::cout << "日照香炉生紫烟，遥看瀑布挂前川。"; //输出语句
5      std::cout << "飞流直下三千尺，疑是银河落九天。"; //输出语句
6      return 0;                //返回0
7  }
```

和 <> 是语法规则，别漏了！



(3)

代码

```
int main() {  
  
    return 0;  
}
```



在C++中，`int main()`是程序的主函数，这是程序执行的起始点。

- `int`是一种数据类型（整型），表示主函数有一个整型的返回值，后面的代码 `return 0;`返回了整型数字0。
- `{ }`表示了主函数的代码块，指令就编写在这对花括号里面。
- `return 0`是一个返回语句，返回0表示程序成功地执行完毕。



敲黑板

主函数有起始点，同时需要结束点，所以发明者设计了`{ }`，成对的符号可以便捷地约束起始点和结束点。

想一想，还有哪些符号是成对的？

`()`、`[ ]`、`" "`、`' '`、`< >`，它们也有大用途，在后面的学习中都会讲到。

(4) `cout << "《望庐山瀑布》"`。

翻译助力理解

- `cout`：这里是“character output”的缩写，意思为字符输出。

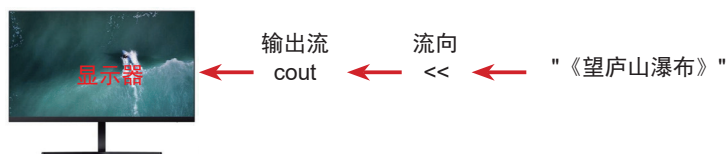
将“《望庐山瀑布》”这个文本传递给输出流对象`cout`，然后显示在屏幕上。“`"`”里面包裹的文本就是要输出的内容。



划重点

`<<`方向是重点

方向代表了流向，`cout << "《望庐山瀑布》"`的流向是输出，所以箭头指向`cout`的方向。



数据流向哪里，方向就朝哪里。

(5) 代码语句的结束标识。

写作时，一句话写完后通常会以句号（。）结尾。而在C++编程中，执行语句以英文格式的分号（**;**）结尾，告诉计算机这句话结束了。



敲黑板

程序执行指令结束都需要用 **;** 结尾。

- **#include <iostream>** 预处理命令是准备动作，所以不用**;**结尾。
- **{ }** 不是实际的命令执行语句，也不用**;**结尾。

► 提出思考

“《望庐山瀑布》日照香炉生紫烟，遥看瀑布挂前川。飞流直下三千尺，疑是银河落九天。”运行结果将标题和诗句排成一排了，怎么分行呢？

只需要加上**endl**即可，它可以让输入结束一行后开启新的一行。

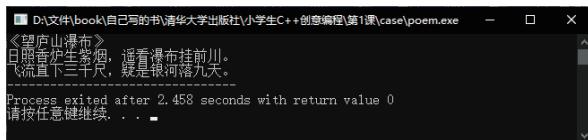
翻译助力理解

- **endl**: 这是end line的意思，表示结束一行。

代码

```
1  #include <iostream>           //头文件
2  using namespace std;         //命名空间
3  int main() {                 //主函数
4      cout << "《望庐山瀑布》" << endl;           //结束一行
5      cout << "日照香炉生紫烟，遥看瀑布挂前川。" << endl; //结束一行
6      cout << "飞流直下三千尺，疑是银河落九天。"; //输出语句
7      return 0;               //返回0
8  }
```

运行程序，标题和诗句分开了。



▼ 捣鼓字符

学会输出后，我们一起来捣鼓一下字符，运用字符输出有趣的图案。



“纸上得来终觉浅，绝知此事要躬行。”编程高手都是敲代码练出来的。

代码

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4
5     cout << "          .+++++. . " << endl;
6     cout << "          * -- . -- . * " << endl;
7     cout << "          * 0 0 * " << endl;
8     cout << "          *   \   * " << endl;
9     cout << "          * WW * " << endl;
10    cout << "          _ooo_ _|_|_ " << endl;
11    cout << " * " << endl;
12    cout << " * 天行健君子以自强不息，地势坤君子以厚德载物 " << endl;
13    cout << " * _ooo_ " << endl;
14    cout << "          | | | " << endl;
15    cout << "          | | | " << endl;
16    cout << "          | _ | _ | " << endl;
17    cout << "          | | | " << endl;
18    cout << "          | _ | _ | " << endl;
19    cout << "          (--) Y (--) " << endl;
20    cout << "          ( ) ( ) ";
21
22    return 0;
23 }
```

运行看看结果：

► 巩固练习

(1) 以下哪个是C++源程序的文件名呢？（ ）

- A. 自我介绍.pptx
- B. 优美音乐.mp3
- C. first.cpp
- D. 一架飞机.jpg

(2) 找出程序中的两处错误，并在代码中改正。

代码

```
include <iostream>
using namespace std;

int main() {

    cout << "请找出代码中的两处错误！"<<endl;
    return 0
}
```

(3) 运用cout输出一架飞机，记得秀一秀正确的程序结果。

代码

```
          #
        #
      ##    ###
    ###    ####
  #####
#####
      ####
      ###
      ##
      #
```

第2课



蹩脚的“ChatGPT” (信息输入)

要和计算机交朋友，就少不了互动，互动就离不开输入输出。不同的输入结合不同的算法得到不同的输出。

计算器中，输入数字，经过四则运算，输出答案。

摄像机中，输入画面，经过剪辑处理，输出视频。

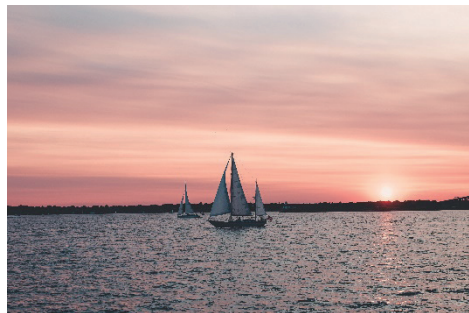
学习机中，输入题目，经过搜索分析，输出题解。

输入 → 算法 → 输出

► 温故知新

现在给你输入一些画面，然后经过你的大脑的思考加工，找到对应画面的李白的诗句，并运用`cout`输出你联想到的诗句吧。

1



2



3



给上面每幅画，编写一段程序输出对应的诗句吧。

(1)

代码

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      cout << "长风破浪会有时，直挂云帆济沧海。";
5      return 0;
6  }
```

(2)

代码

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      cout << "举头望明月，低头思故乡。";
5      return 0;
6  }
```

(3)

代码

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      cout << "两岸猿声啼不住，轻舟已过万重山。";
5      return 0;
6  }
```

▼ 创造我的“ChatGPT”

ChatGPT是人工智能技术驱动的自然语言处理工具，它能够基于在预训练阶段所见的模式和统计规律生成回答，并且能根据聊天的上下文进行互动，实现像人类一样的聊天交流。此外，它甚至能完成撰写邮件、视频脚本、文案、翻译、代码以及写论文等任务。



我就是这样，敢想敢为，想到了就去创造。

ChatGPT太强大了，我非常崇拜它。虽然我只学了几天的C++，但是我决定创建一个。

于是，我成功地创造了一个调皮的“ChatGPT”。

代码

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main() {
6     string question;
7     cout << "我是蹩脚的ChatGPT，你有什么问题吗？" << endl;
8     cin >> question;
9     cout << "都说我是蹩脚的ChatGPT，所以我不知道[" + question + " ]的答案。" ;
10    return 0;
11 }
```

运行程序后，它问我：

我是蹩脚的ChatGPT，你有什么问题吗？

我提出问题：

怎样才能成为C++大神？

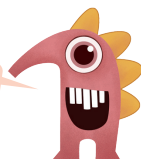
它回答道：

都说我是蹩脚的ChatGPT，所以我不知道【怎样才能成为C++大神？】的答案。

运行效果是这样的：

```
D:\文件\book\自己写的书\清华大学出版社\小学生C++创意编程\第2课\case\蹩脚的ChatGPT.exe
我是蹩脚的ChatGPT，你有什么问题吗？
怎样才能成为C++大神？
都说我是蹩脚的ChatGPT，所以我不知道【怎样才能成为C++大神？】的答案。
Process exited after 15.38 seconds with return value 0
请按任意键继续. . .
```

这回答，好气又好笑。



还真是蹩脚的程序，不过它竟然能知道我问的问题，看来它还是有两把刷子。让我们一起探索一下程序是如何知道我们提出问题的。

(1) 要使用 `string` 工具，则先要将头文件包含进程序，使用预处理命令 `#include <string>`。

(2) `string question` 表示声明一个名为 `question` 的变量，该变量的类型是 `string`。



敲黑板

想象一下，在计算机中，你创建了一个魔法盒子，可以用来存放各种东西。这个盒子上贴着一个特殊的标签，叫作“变量”。当我们需要往计算机里存放东西时，就去创建一个魔法盒子。

举一个例子：现在我创建了一个 `question` 的魔法盒子（变量），把我输入的问题存放在里面。当我或者计算机想要知道里面的问题时，只需要找到 `question` 就可以知道里面存放的内容。

在计算机中，将东西放入魔法盒子（变量），通常是称为 **赋值**。

(3) `cin >> question` 将输入的内容 **赋值** 给 `question` 变量。这时候魔法盒子 `question` 里面存放的就是输入的内容。

翻译助力理解

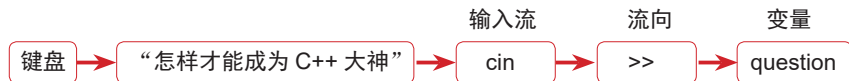
- `cin`：是 `console input` 的缩写，表示从控制台输入数据。



划重点

`>>` 方向是重点

方向代表了流向，`cin >> question` 的流向是从输入流向变量。



对比学习 `cout <<`

(4) `cout << "都说我是蹩脚的ChatGPT，所以我不知道[" + question + "]" 的答案。"`。

因为将输入的问题赋值给了 `question`，所以这里使用该变量就可以知道问题的内容了。虽然蹩脚，但是程序将回答的话语和问题用 `+` 进行了组合。

在 C++ 中，放在双引号（`" "`）里面的内容被称为 **字符串**，这样就能原样输出了。

如果是这样，编译器就会把 `question` 当作一个英文单词直接输出，而不是当作输出变



量question中的内容。

`cout << "都说了我是蹩脚的ChatGPT，所以我不知道[question]的答案。"`

为了避免这个问题，伟大的发明者通过**拆分**再**组合**的方式实现了输出。将输出内容拆成3部分后通过+进行组合。

①"都说了我是蹩脚的ChatGPT，所以我不知道["

+

②question

+

③"]的答案。"



原来有些问答就是这样来的呀！

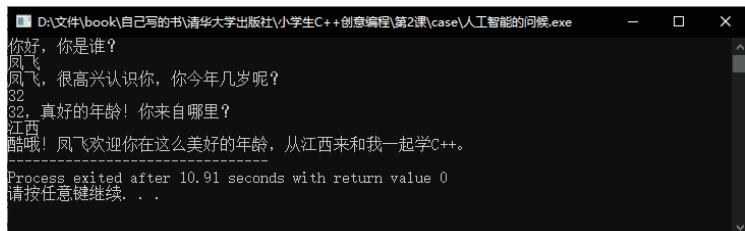
▼ 人工智能的问候

试着与计算机对话，它可以记住你的名字、家乡、年龄等。

代码

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main() {
6
7      string name, age, hometown;
8
9      cout << "你好，你是谁？" << endl;
10     cin >> name;
11     cout << name + "，很高兴认识你，你今年几岁呢？" << endl;
12     cin >> age;
13     cout << age + "，真好的年龄！你来自哪里？" << endl;
14     cin >> hometown;
15     cout << "酷哦！" + name + "欢迎你在这么美好的年龄，从" + hometown + "
    来和我一起学C++。";
16
17     return 0;
18 }
```

运行看看结果：



► 巩固练习

(1) 哪段代码可以将键盘输入的内容赋值给变量word? ()

- A. `cout >> word`
- B. `cin << word`
- C. `cout << word`
- D. `cin >> word`

(2) 以下代码可以输出“杨梅和葡萄真好吃!”。() ✓ () ×

代码

```
#include <iostream>
#include <string>
using namespace std;

int main() {

    string waxberry, grape;

    cout << "输入你喜欢吃的水果。" << endl;
    cin >> waxberry;
    cout << "还有其他水果吗?" << endl;
    cin >> grape;
    cout << waxberry + "和grape真好吃! ";

    return 0;
}
```

(3) 脑洞大开，制作一段有趣的问答程序。