集训课-01：初识C++

课程目标

01 Dev c++ 的安装及使用

02 输出语句

03 算术运算符

04 /10和%10的妙用

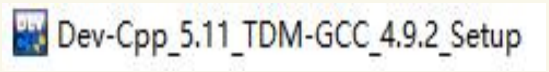
05 总结回顾

Part 1

Dev c++ 的安装及使用

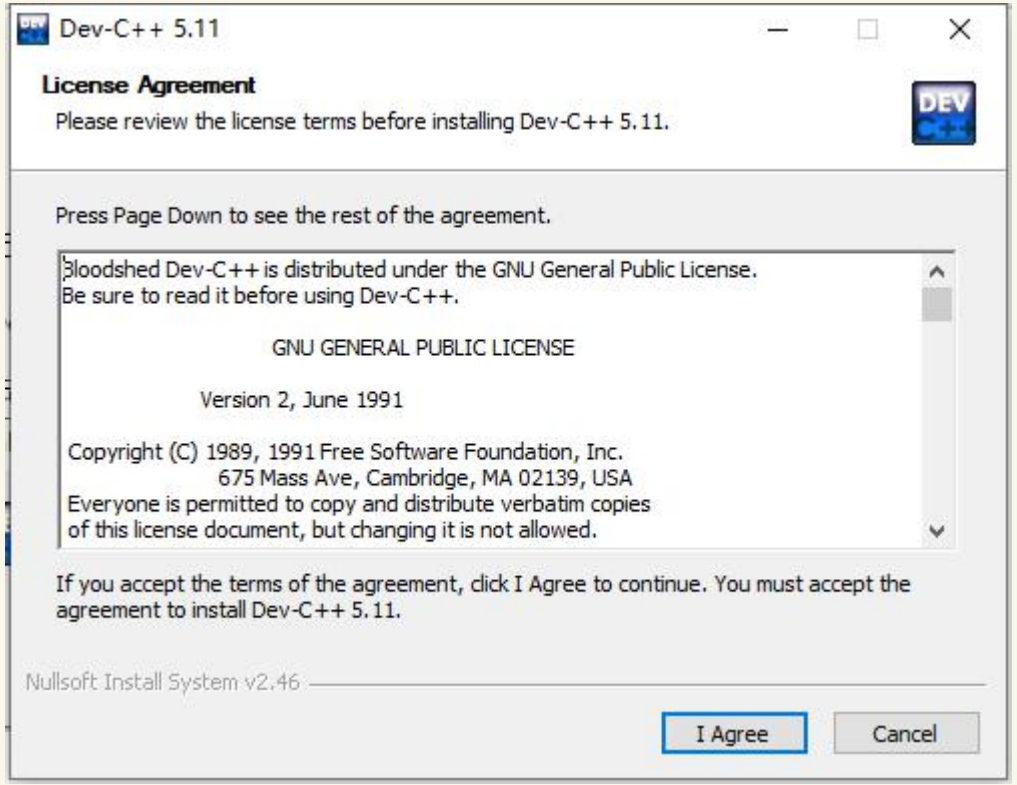
Dev C++的安装

1、双击运行此程序



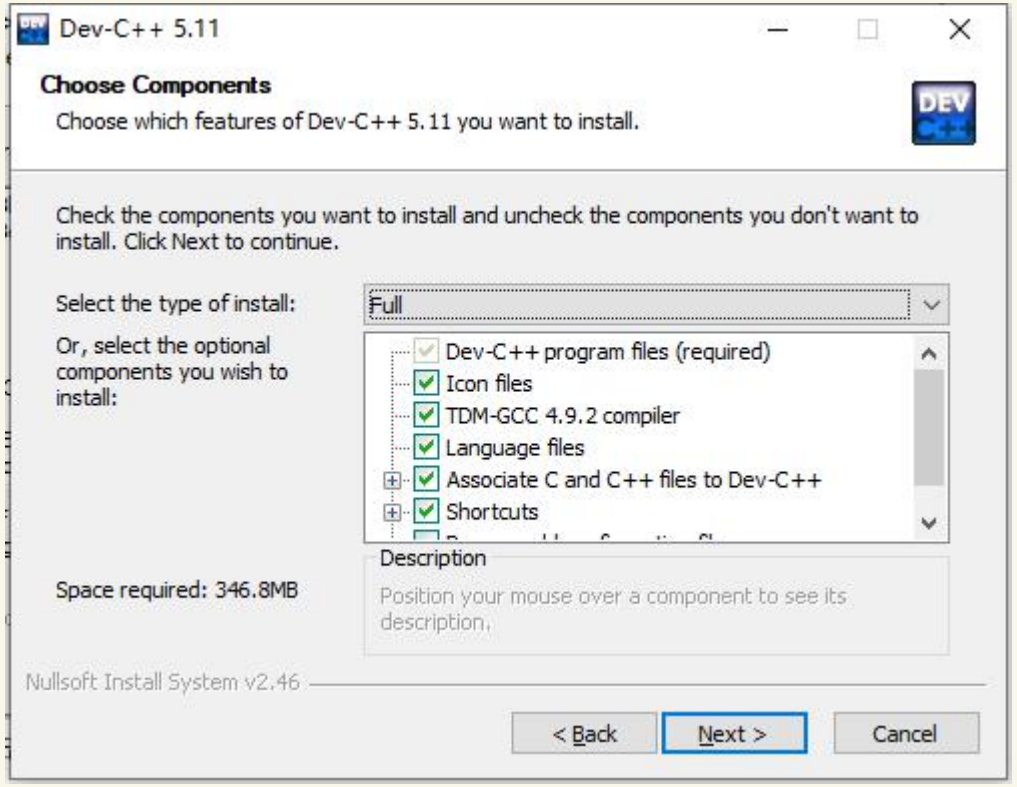
2、点击OK

3、点击I Agree

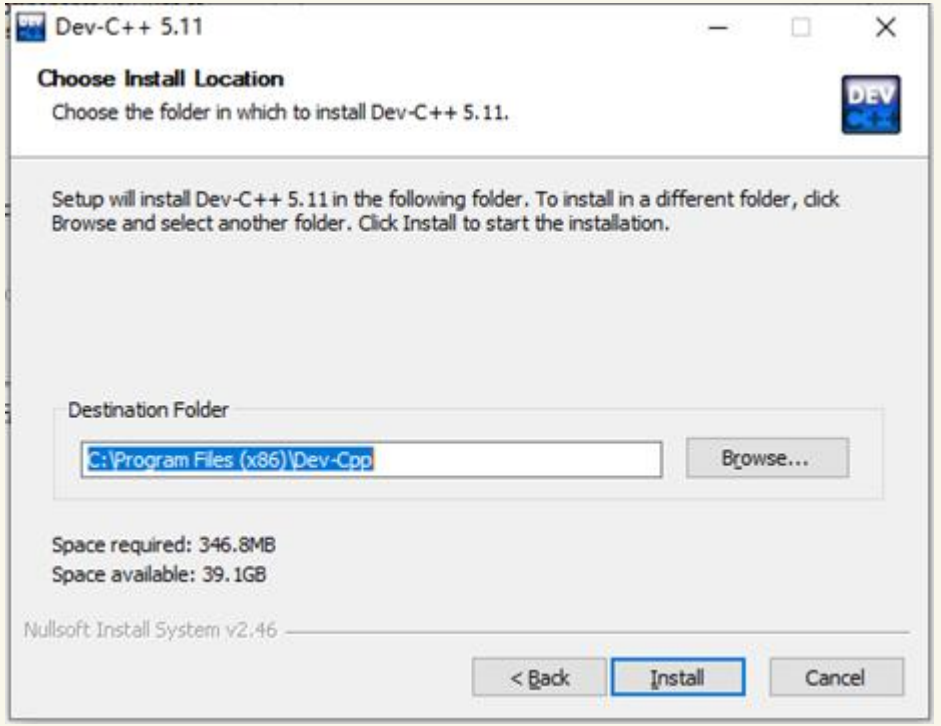


Dev C++的安装

4、点击Next

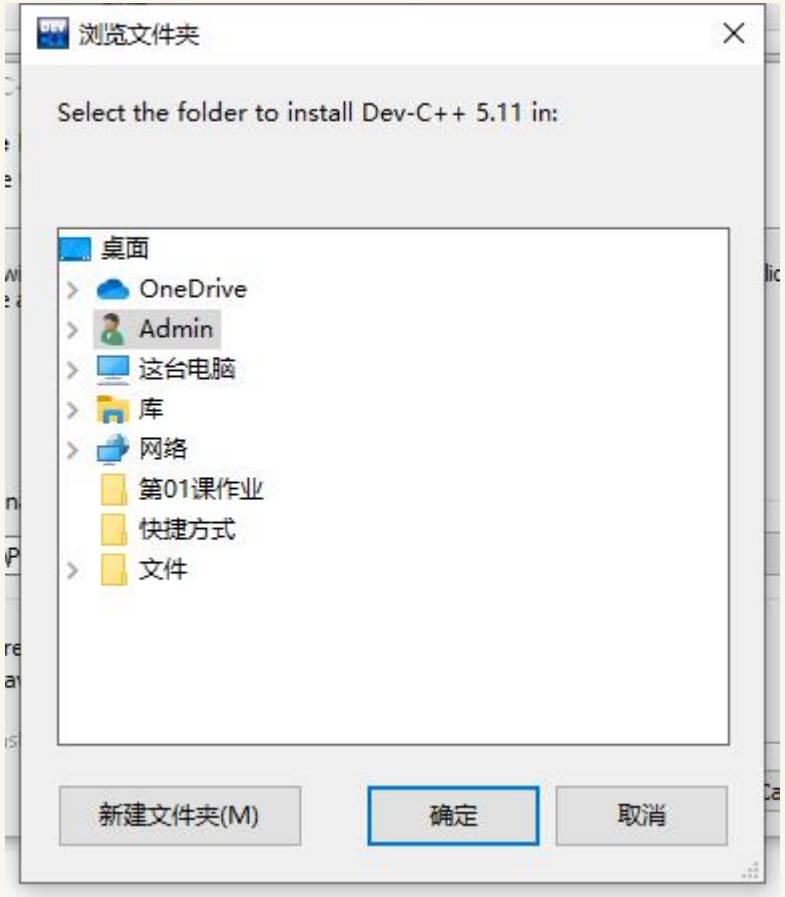


5、点击Browse

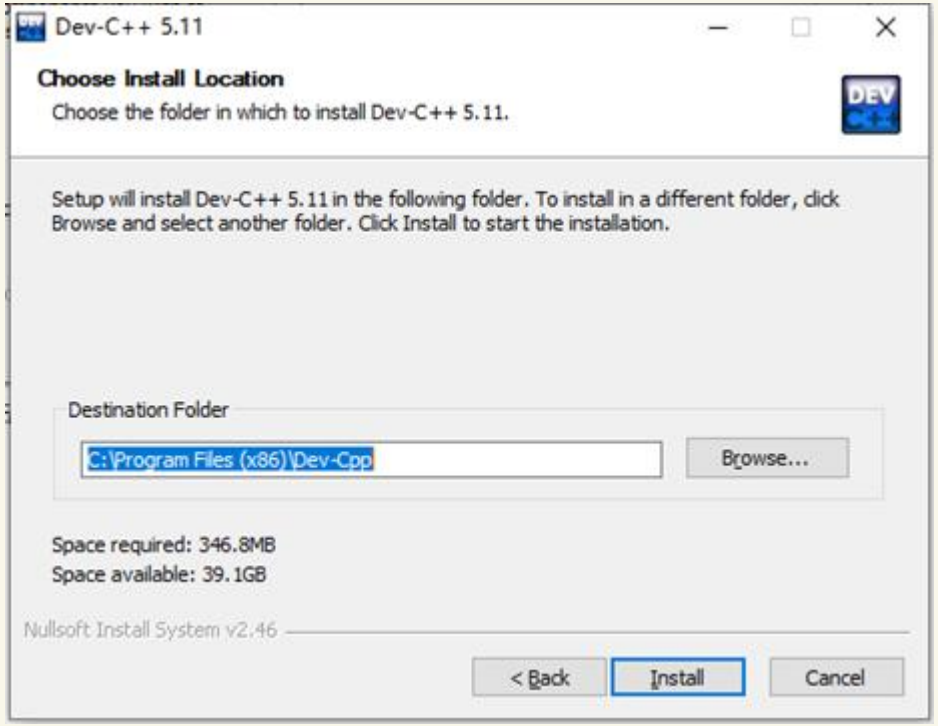


Dev C++的安装

6、选择安装路径后点击确定

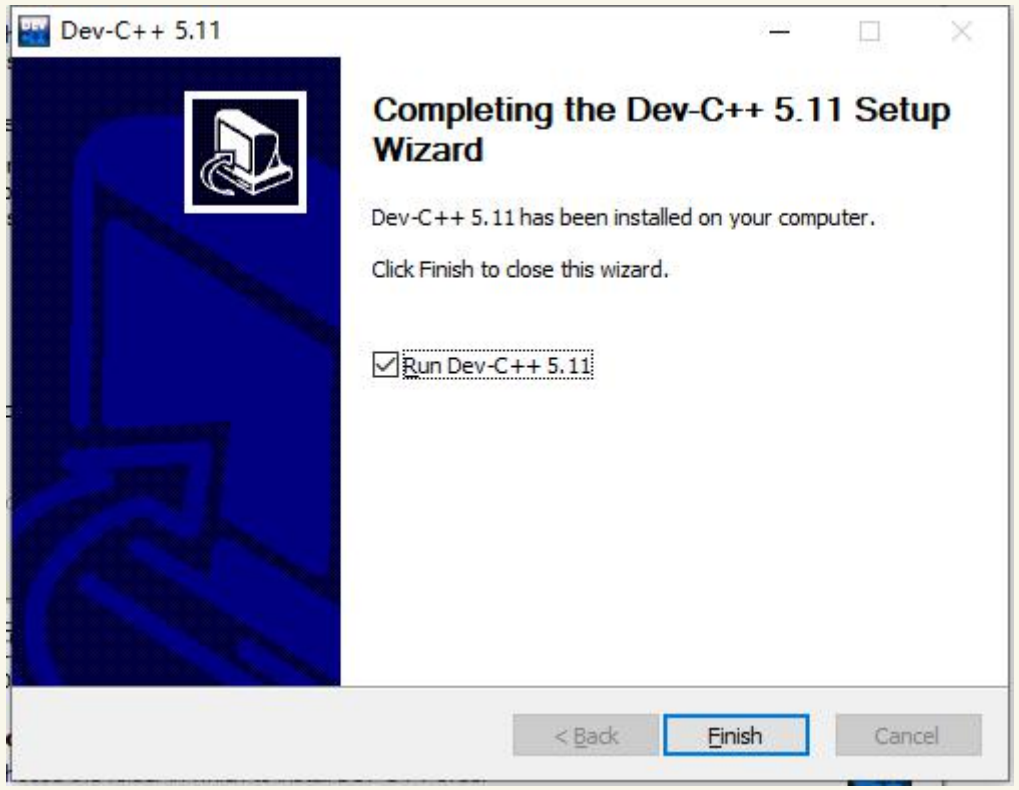


7、点击Install安装



Dev C++的安装

8、点击Finish

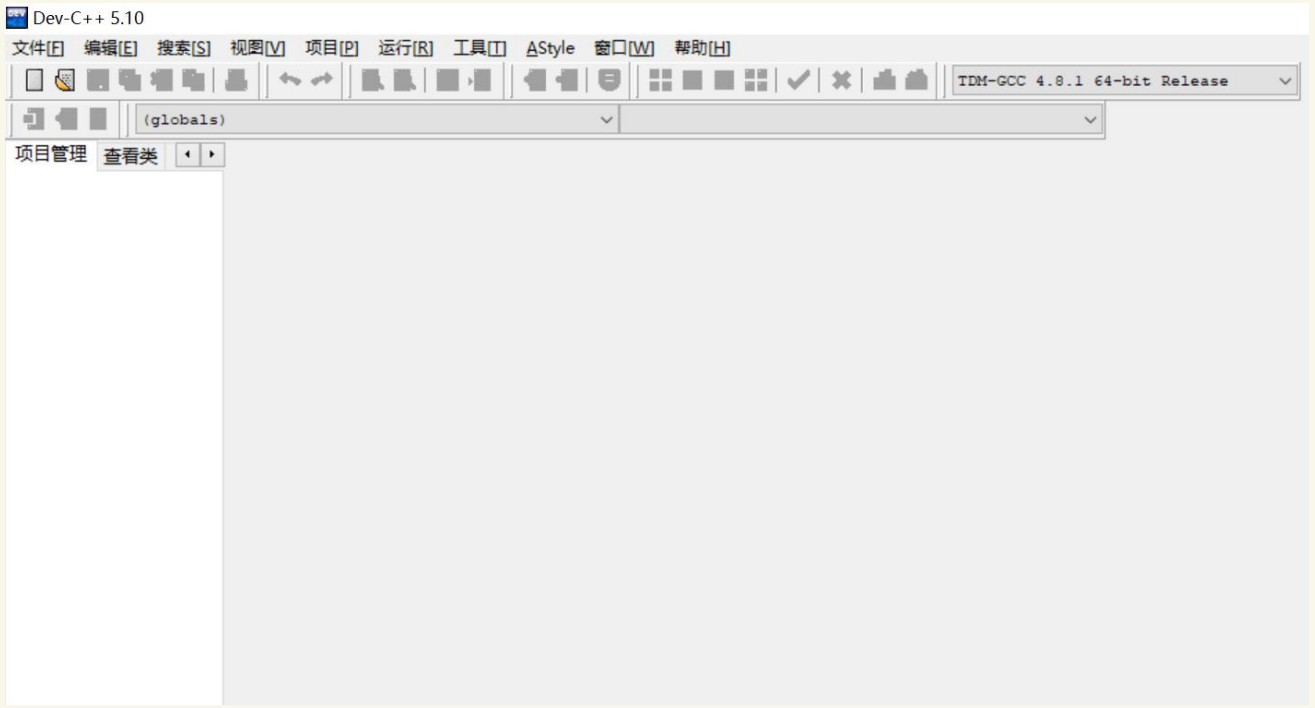


Dev C++的使用

1、创建新文件

方式一：点击文件下方的图标，再点击源代码

方式二：按住ctrl+n创建

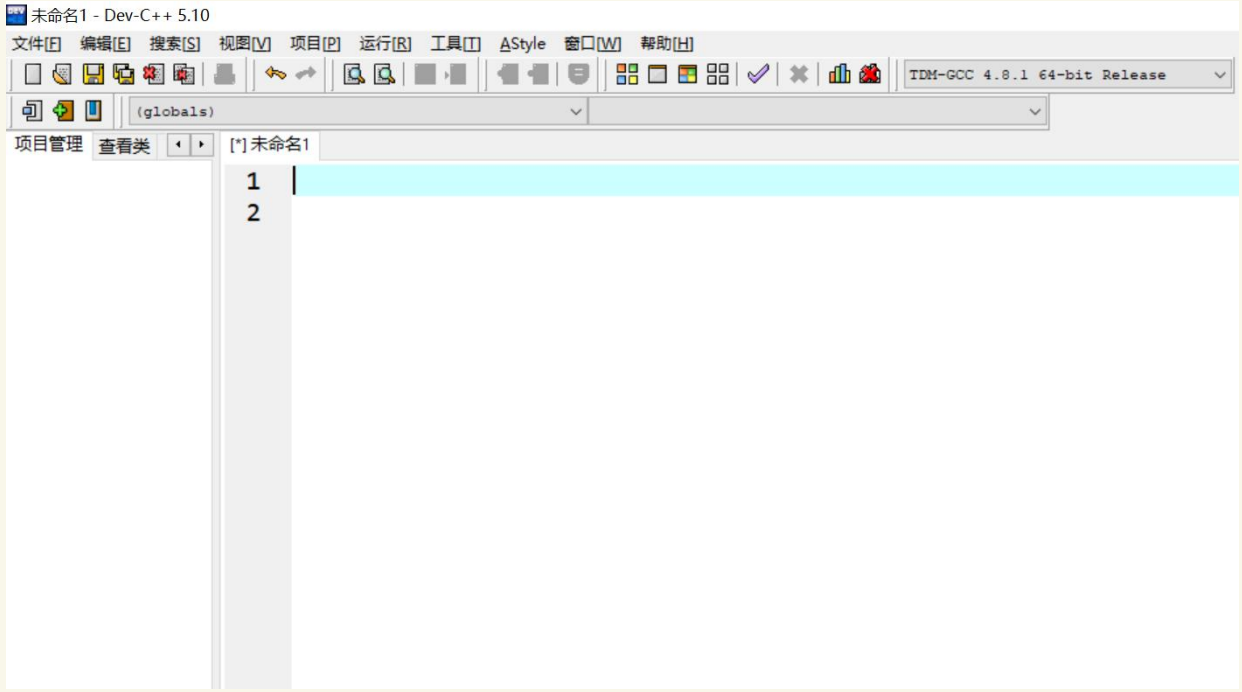


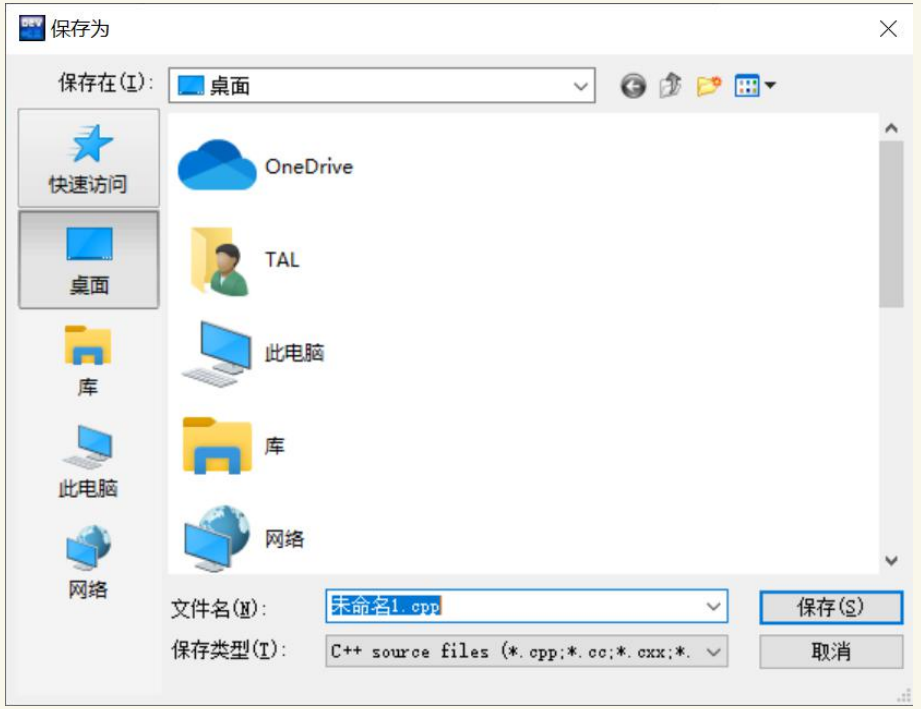
Dev C++的使用

2、新文件的保存

方式一：点击文件，再点击保存，弹出窗口输入文件名选择保存位置后，点击保存

方式二：按住ctrl+s进行保存，弹出窗口输入文件名选择保存位置后，点击保存



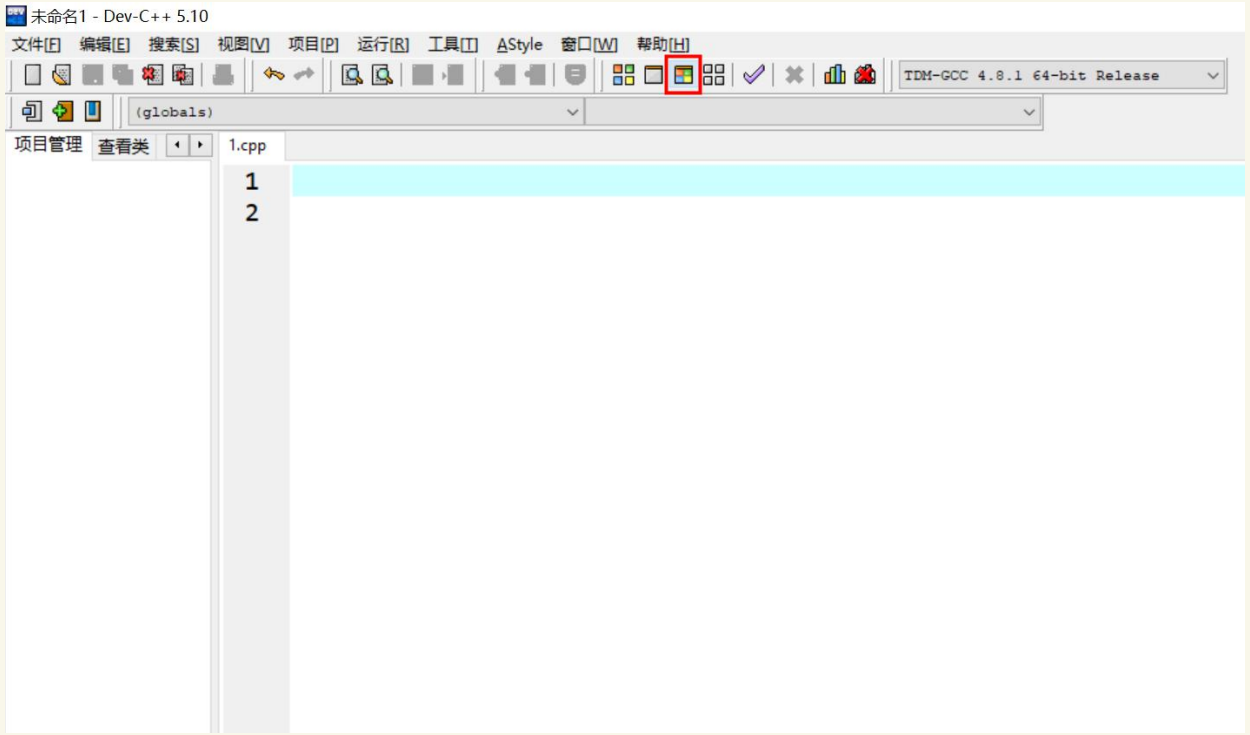


Dev C++的使用

3、运行程序

方式一：点击如下图标运行

方式二：按下键盘上F11运行（部分笔记本电脑需要Fn+F11运



Dev C++的使用

4、字体放大缩小

Ctrl+鼠标滚轮向上：放大字体

Ctrl+鼠标滚轮向下：缩小字体

Dev C++的使用

5、C++程序基本框架

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

int main()
{

    return 0;
}
```

Dev C++的使用

5、C++程序基本框架

预处理

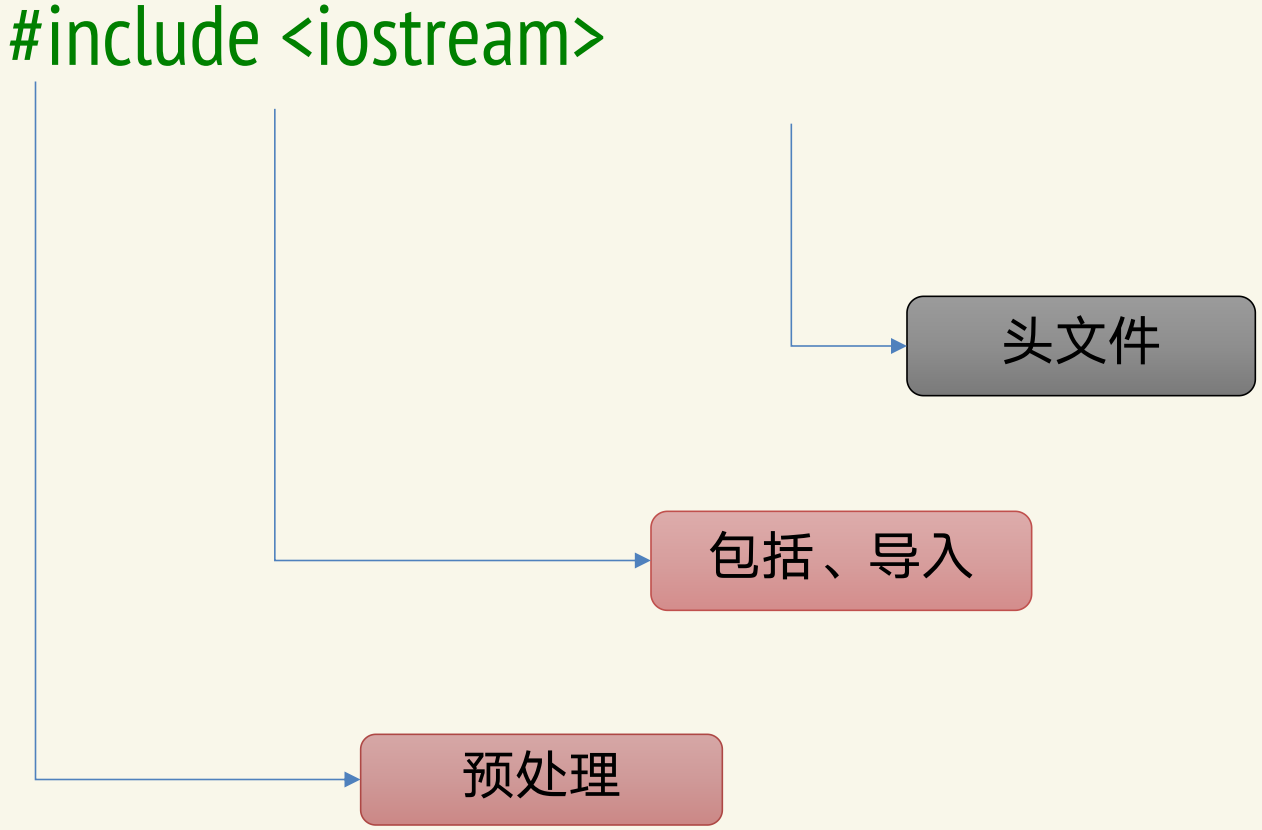
主函数

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "Nice to meet you.";
7      return 0;
8  }
```

一个程序只有一个主函数

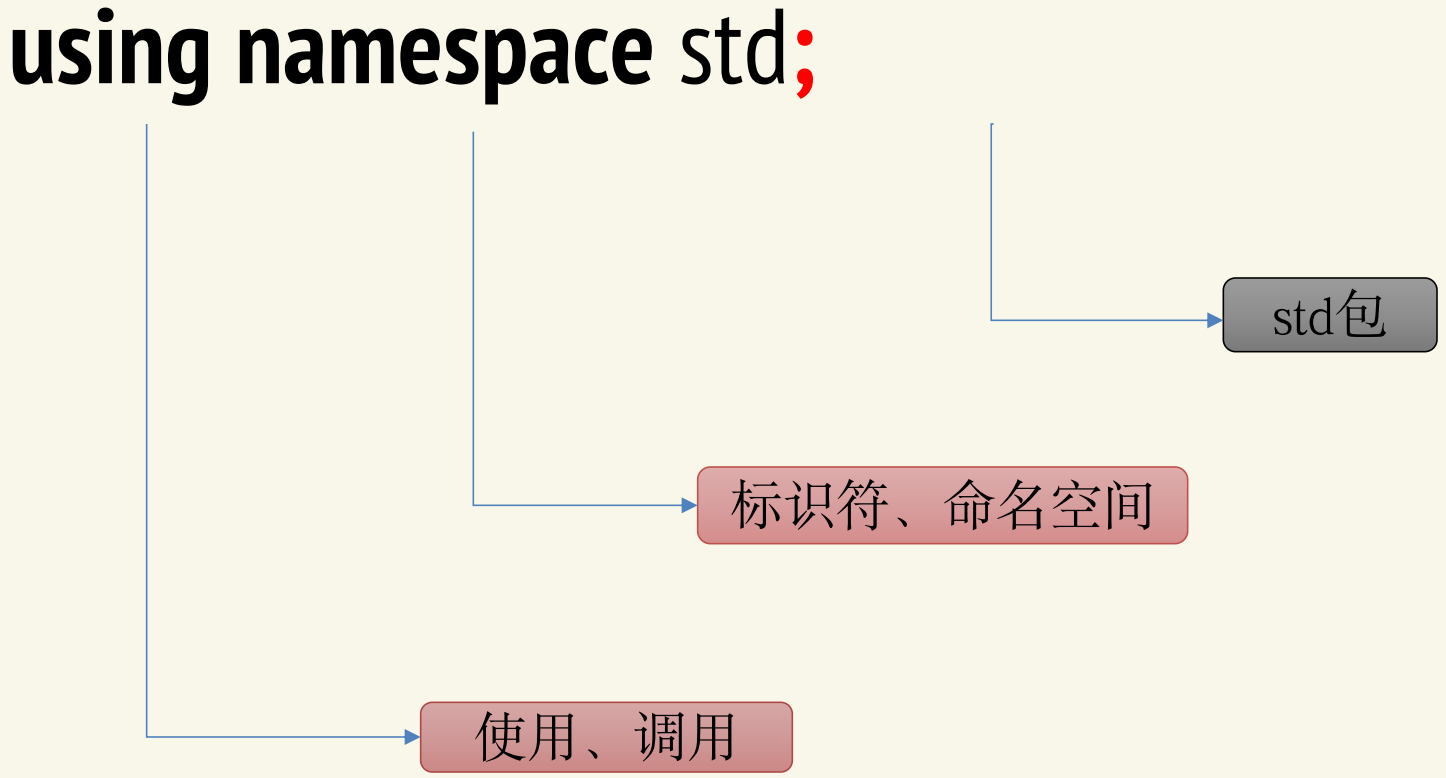
Dev C++的使用

5、C++程序基本框架



Dev C++的使用

5、C++程序基本框架



Dev C++的使用

5、C++程序基本框架

```
int main()
{

    return 0;
}
```

程序入口

程序出口/结束程序

Dev C++的使用

5、C++程序基本框架

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

int main()
{

    return 0;
}
```

C++程序的框架由头文件，命名空间，主函数三部分构成

Dev C++的使用

想一想1：

以下说法有误的是

A.一个C++程序只能有一个主函数

B.一个C++程序可以有多个主函数

C.一个C++程序的框架由头文件，命名空间，主函数三部分构成

Part 2

输出语句

输出语句

```
cout <<" 需要输出的内容 ";
```

cout 输出流

输出语句

```
cout <<" 需要输出的内容 ";
```

<< 输出符号
必须搭配cout使用

输出语句

```
cout << " 需要输出的内容 ";
```

" "　放在双引号里面的内容叫　字符串常量
放在双引号里面的内容会被　原样输出

输出语句

```
cout<<"需要输出的内容";
```

; 每行程序以分号结尾

输出语句

```
cout <<" 需要输出的内容 ";
```

cout　输出流

<<　输出符号，必须搭配cout使用

" "　放在双引号里面的内容叫　字符串常量
放在双引号里面的内容会被　原样输出

;　每行程序以分号结尾

输出语句

想一想2：

下列说法不正确的是

A、cout必须搭配<<符号使用

B、放在双引号里面的内容会被原样输出

C、每行程序要以逗号结尾

输出语句

- 使用 << 连接多个内容

```
cout << "项目1" << "项目2" << "项目n";
```

- 使用多个cout语句实现

```
cout << " 项目1 ";
cout << " 项目2 ";
cout << " 项目n ";
```

输出语句

换行

endl = end + l (line)

注意：

endl前面必须要有<<

例如：

```
cout << "爱你孤身走暗巷" << endl;
cout << "爱你不跪的模样" << endl;
```

输出语句

想一想3：

下列哪个代码可以实现换行输出

A、cout << "hello" << end;

cout << "123";

B、cout << "good luck" << endl;

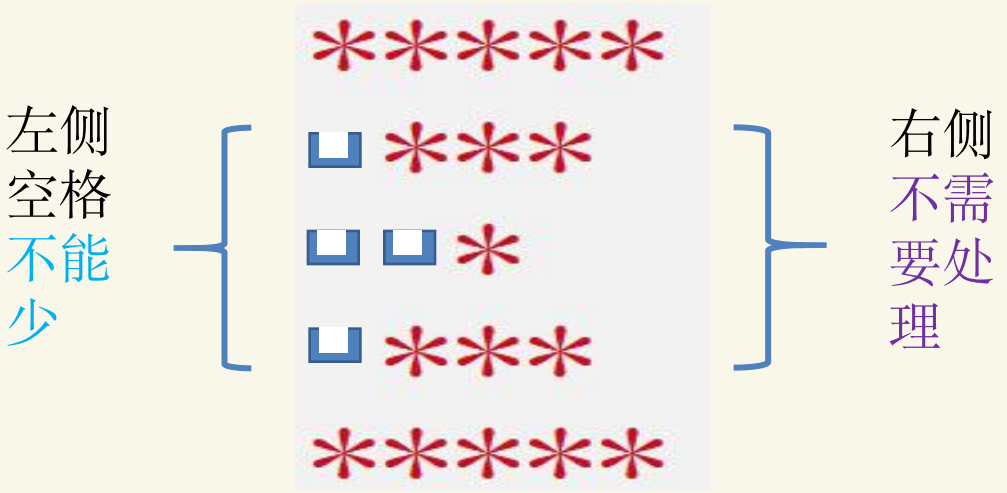
cout << "to you";

C、cout << "good luck" endl;

cout << "to you";

输出语句

- 顺序结构：
  按照从上到下、从左到右依次执行的程序结构



输出语句

想一想4：

下列关于顺序结构说法正确的是

A、按照从上到下、从左到右依次执行的程序结构

B、按照从下到上、从右到左依次执行的程序结构

输出语句

一、基本知识

1、 cout 输出流：将后面的内容显示到终端
2、 << 输出符号，和cout搭配使用
3、 " " 放在双引号里面的内容会被原样输出
4、 ; 程序以分号结尾
5、 endl 实现换行，前面必须要有<<

二、组合输出

1、cout << 项目1 << 项目2 << 项目n; //输出内容在同一行
2、cout << 项目1;
cout << 项目2; //输出内容在同一行，换行需要添加endl
cout << 项目n;

三、顺序结构

按照从上到下，从左到右依次执行的程序结构

Part 3

算术运算符

算术运算符

观察右侧第6、7、8行程序，有什么异同？

你得出什么结论？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "10+3" << endl;
7      cout << 10+3 << endl;
8      cout << 13 << endl;
9      return 0;
10 }
```

算术运算符

- cout << "10+3" << endl;
  算术表达式放在双引号里,10+3被原样输出

- cout << 10 + 3 << endl;
  输出算术表达式的结果13

- cout << 13 << endl;
  输出数值13的值

算术运算符

算术运算	数学中的符号	程序中的符号
加	+	+
减	-	-
乘	x	*
除	÷	/
求余数	÷ 搭配……	%（求模）

*　/　% 优先级高于　+　-

算术运算符

想一想5：

算术表达式5 x 3 － 9 ÷ 3转换为程序中的算术表达式，下列选项正确的是

A、5 * 3 － 9 ÷ 3

B、5 x 3 － 9 / 3

C、5 * 3 － 9 / 3

算术运算符

算术运算	数学中	程序中
除	10 ÷ 3 = 3.33333	10 / 3 = 3

两个整数进行 / 运算，结果一定是整数

算术运算符

想一想6：

程序中算术表达式100 / 33的结果正确的是

A、 3

B、 1

C、 3.33333

Part 4

/10和%10的妙用

/10和%10的妙用

- 任何一个整数都可以通过/10去掉个位

/10和%10的妙用

- 任何一个整数都可以通过%10得到个位

/10和%10的妙用

利用/10和%10的结论

如何得到789的百位、十位、个位

/10和%10的妙用

对789进行数位分离

个位：

- ✔ 789 % 10得到数字9

/10和%10的妙用

对789进行数位分离

百位：

- ✔ 789 / 10去掉个位数字9，得到78
- ✔ 78 / 10去掉个位数字8，得到7
- ✔ 789 / 10 / 10等价于789 / 100

/10和%10的妙用

对789进行数位分离

十位：

- ✔ 789 / 10去掉个位数字9后,得到78
- ✔ 78再 % 10得到数字8

/10和%10的妙用

对789进行数位分离

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

int main()
{
        cout << 789 / 100 << endl;
        cout << 789 / 10 % 10 << endl;
        cout << 789 % 10;
        return 0;
}
```

/10和%10的妙用

- 在数学算术表达式中，可以有大括号，中括号，小括号，在程序中只有小括号

- 数学中：5x{13+[1024÷10-(2+3)]}÷2

- 程序中：5 * (13 + (1024 / 10 - (2 + 3))) / 2

/10和%10的妙用

- 注释：
  对程序的解释、提示说明，不参与程序实际运行,不能把参与运行的内容注释掉

- //　　单行注释符号

- /*　　*/　　多行注释符号

Part 5

总结回顾

总结回顾

一、基本知识

1、 cout　　输出流：将后面的内容显示到终端
2、 <<　　输出符号，和cout搭配使用
3、 " "　　放在双引号里面的内容会被原样输出
4、 ;　　程序以分号结尾
5、 endl　　实现换行，前面必须要有<<

二、组合输出

1、cout<<项目1<<项目2<<项目n;　　//输出内容在同一行
2、cout<<项目1;
　　cout<<项目2; //输出内容在同一行，换行需要添加endl
　　cout<<项目n;

三、顺序结构

按照从左到右，从上到下依次执行的程序结构

总结回顾

1、算术运算算符：+ - * / %

2、两个整数（整型数据）进行/运算得到的结果一定是整数

3、任何一个整数都可以通过/10去掉个位数字

4、任何一个整数都可以通过%10得到个位数字

5、程序中的算术表达中只能使用圆括号

6、//单行注释 /* */多行注释， 被注释的内容不参与程序运行

Class is over