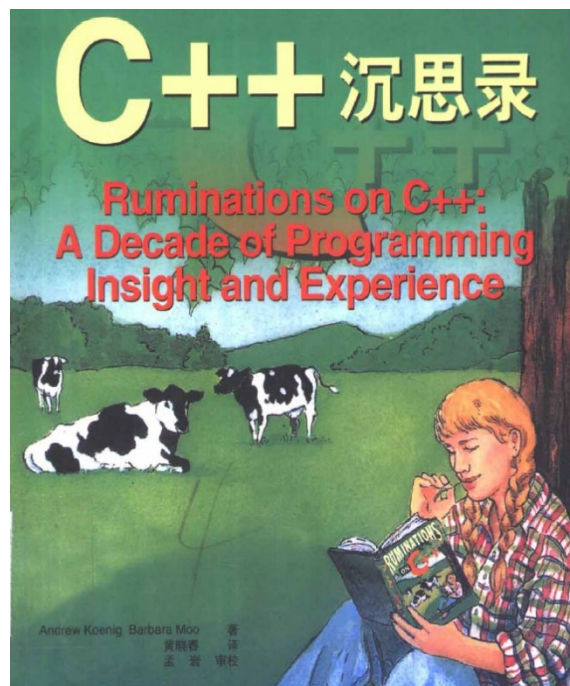


# 高级语言C++程序设计

## Lecture 1 课程简介



# 1.1 这是一门什么课?



# 高级语言C++程序设计

---

**程序：** 让计算机完成某种任务的一组指令



# 高级语言C++程序设计

---

语言：交流沟通的工具



# 高级语言C++程序设计

---

程序语言：人与计算机“沟通”的工具

已知a为5, b为4, c  
为a和b的和, 计算c



```
int a = 5;  
int b = 4;  
int c = a + b;
```

程序



# 高级语言C++程序设计

## 偏向人类语言的程序语言

已知a的值为5, b  
的值为4, 计算a+b



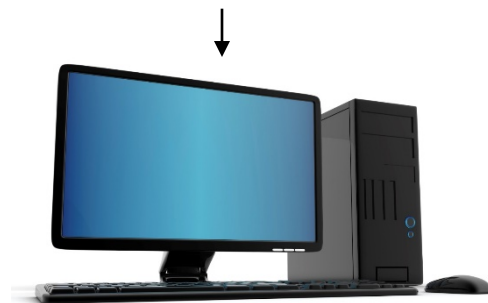
```
int a = 5;  
int b = 4;  
int c = a + b;
```

程序

“翻译”

机器指令

```
11111111 11011000 11111111 11100000 00000000  
00010000 01001010 01000110 01001001 01000110  
00000000 00000001 00000010 00000000 00000000  
00000100 00000000 00000011 00000000 00000000  
11111111 11111110 00000000 00010000 01001100  
01100001 01110110 01100011 00110101 00110111  
00101110 00111000 00111001 00101110 00110001  
00110000 00110000 00000000 11111111 11011011  
00000000 01000011 00000000 00001000 00000100  
00000100 00000100 00000100 00000100 00000101  
00000101 00000101 00000101 00000101 00000101  
00000110 00000110 00000110 00000110 00000110  
00000110 00000110 00000110 00000110 00000110  
00000110 00000110 00000110 00000111 00000111
```

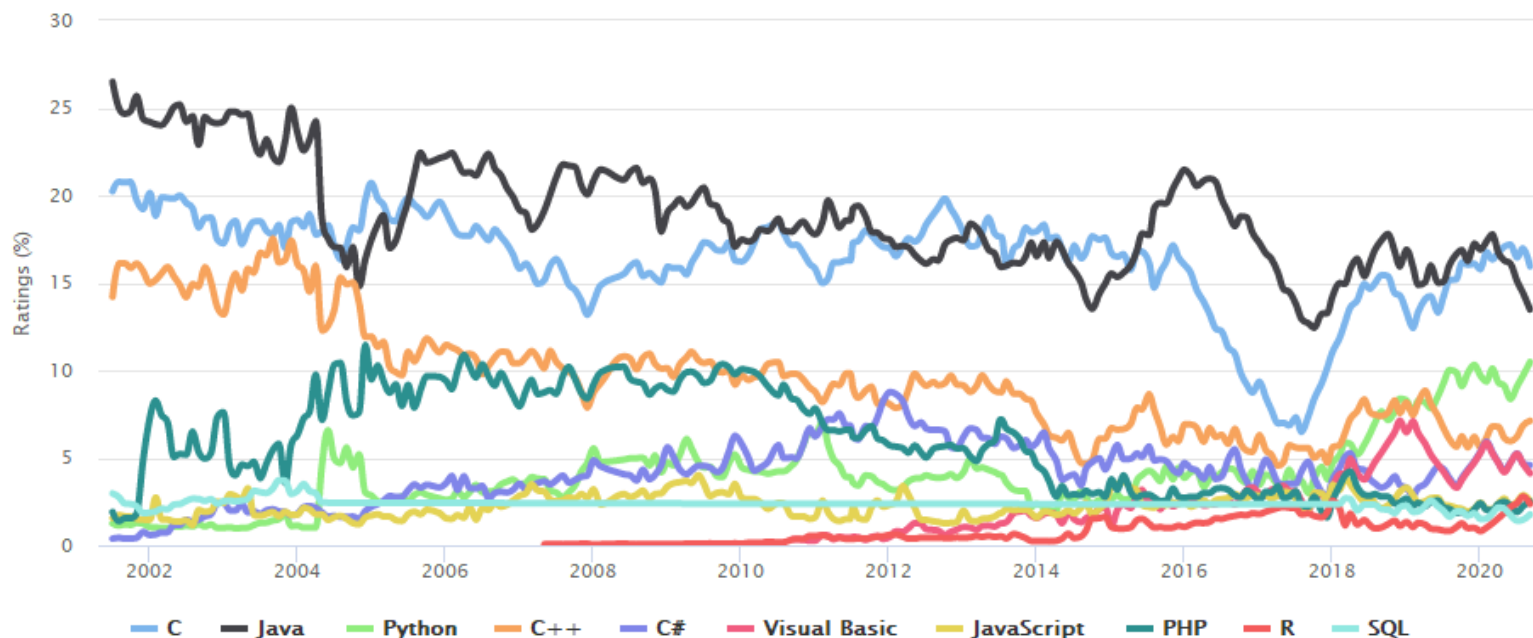


# 高级语言C++程序设计

C++：一种非常强大的高级程序设计语言

TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)





# 高级语言C++程序设计

---

**B语言(Basic Combined Programming Language)**是贝尔实验室开发的一种通用的程序设计语言，它是于1969年前后由美国贝尔实验室的电脑科学家肯·汤普逊 (Ken Thompson) 在丹尼斯·里奇 (Dennis Ritchie) 的支持下设计出来

1972年，美国贝尔实验室的 D.M.Ritchie 在B语言的基础上最终设计出了一种新的语言，他取了BCPL的第二个字母作为这种语言的名字，**这就是C语言**。

1979年，Bjame Sgoustrup到了Bell实验室，开始从事将C改良为带类的C (*C with classes*) 的工作。1983年该语言被正式命名为**C++**。

1989年, C++2.0版  
1993年, C++3.0版  
1998年, C++标准诞生  
2011年, C++11标准  
2014年, C++14标准  
2017年, C++17标准

---



# 高级语言C++程序设计

---

## 用高级程序语言C++设计并实现程序

作家通过**人类语言**编写著作，向读者传递信息

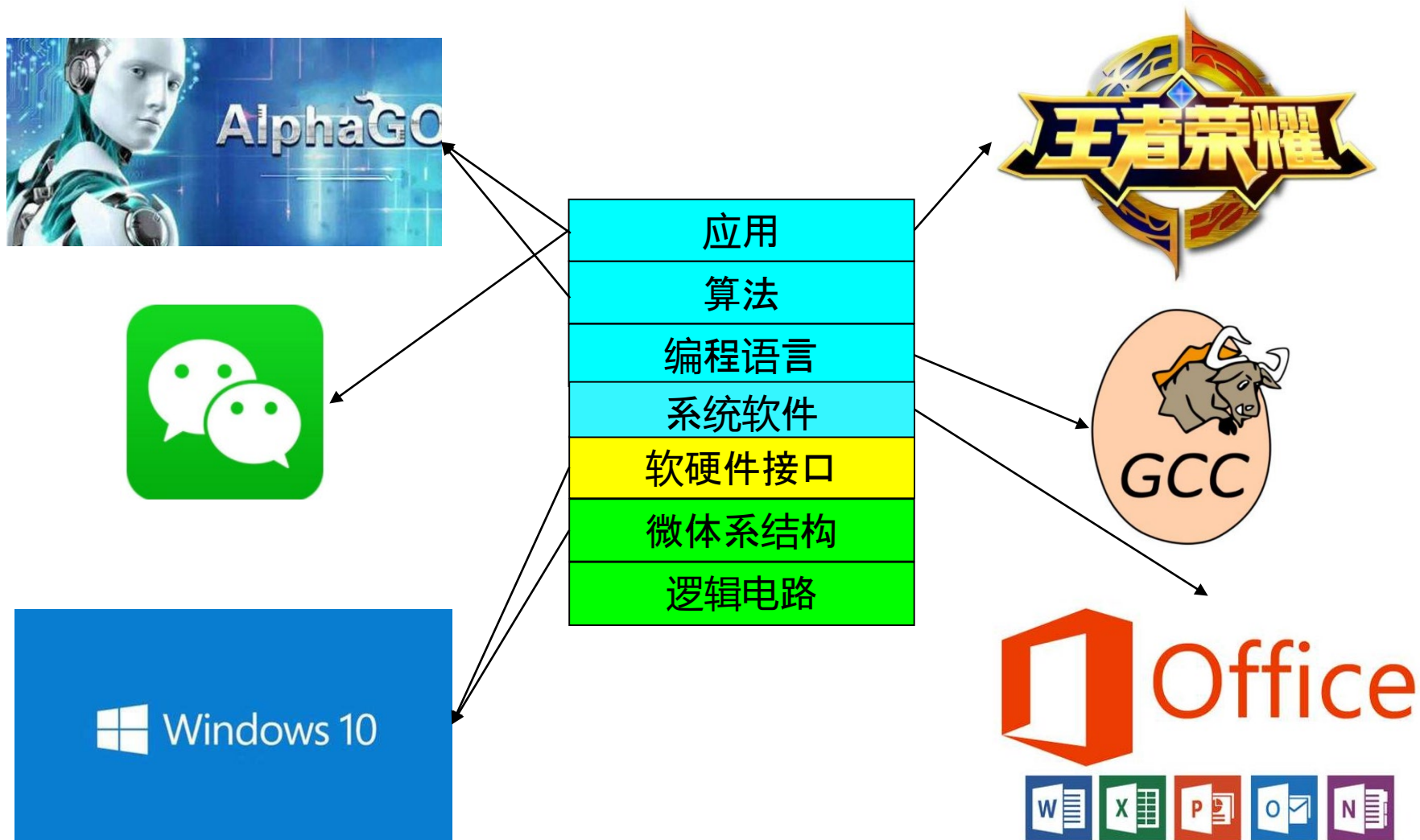


程序员通过**程序语言**编写程序，向计算机传达指令

## 1.2 为什么是C++?

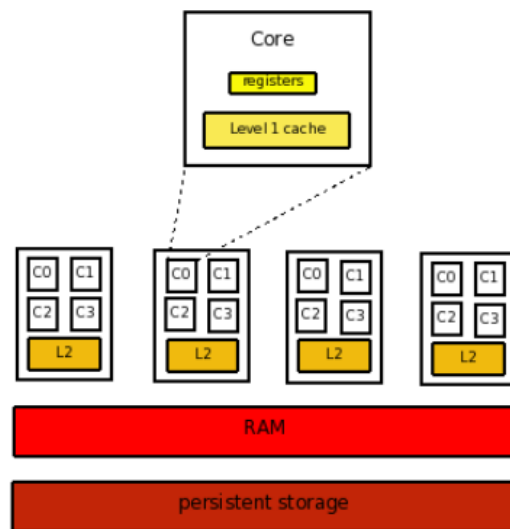


# C++几乎无所不能



# C++是最复杂的编程语言

# C++与计算机结构密切相关，是学好后续课程的重要基础



# 学好C++，走遍天下都不怕!



# C++的其他优点

---

## ■ 为什么选择C++语言

- 面向对象程序设计正在逐渐成为主流设计技术
- OOP 技术并不取代SP 和一般的程序设计的技能技巧
- 由于各大公司的竞相开发，C++语言在各种不同机型上都有优秀的编译系统和相关的环境与工具
- C++语言最可能取代C 而成为主流的软件开发语言之一
- C++语言已成为计算机专业主要的教学语言

现在理解这些还太早！

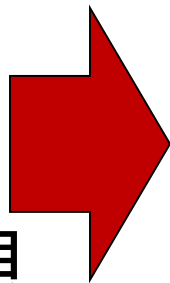
## 1.3 如何学习C++?



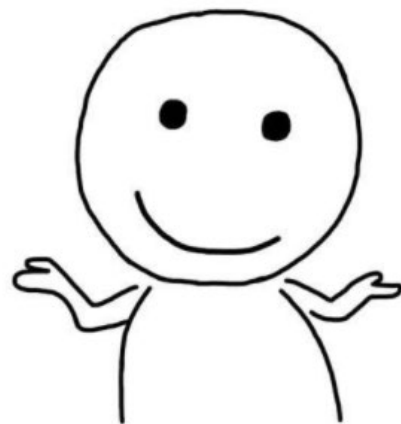
# 学好C++的几点建议

---

- 熟练掌握C++基本语法
  - 上课、教材、网络
- 疯狂上机练习!!!
  - 经典算法倒背如流
- 深入理解C++设计思想
  - 假设你是C++作者
- 敢于挑战经典
  - C++其实存在很多问题



~~我爱英语~~ 编程  
编程 ~~英语使我快乐~~





## 1.4 课程相关

# 教学团队

---



## ■ 主讲 李雨森

- 副教授 @ 南开大学 计算机学院 百度联合实验室
- 博士, 新加坡南洋理工大学
- <https://liyusen-nku.github.io/>
- [liyusen@nbjl.nankai.edu.cn](mailto:liyusen@nbjl.nankai.edu.cn)
- 研究方向：并行与分布式系统，计算机体系结构

## 助教



时浩铭  
2019级卓越班



崔家贺  
2020级软件



李奕琛  
2020级研究生



张自强  
2020级研究生

# 课程安排

---

## ■ 课时安排

- 每周讲授3课时，实验4课时

## ■ 成绩

- 期末笔试成绩 (50%)
  - 上机考试成绩 (30%)
  - 平时成绩 (包括上机作业、测验等, 共20%)
-

# 交流平台

---

## □ 微信群



1051高级语言 2-1 (李雨森)



该二维码7天内(9月24日前)有效, 重新进入将更新

发布通知、课件  
、讨论交流等

## □ 雨课堂

- 微信小程序

- 网页版

(<https://www.yuketang.cn/>)

雨课堂



发布通知、课件  
、课堂测验等

# 参考资料

- 机械工业出版社, C++17入门经典
- 机械工业出版社, The C++ Programming Language
- C++程序设计, 刘璟编著, 高等教育出版社



# 上机平台

---

- 机房

- 编程环境是Visual Studio 2010
- 上机练习、作业、考试

- 南开大学OJ系统

- 上机练习、作业、考试

- OpenJudge

- <http://noi.openjudge.cn/>
- 进阶练习

- 其他编程练习网站

- 洛谷(<https://www.luogu.com.cn/>)
-

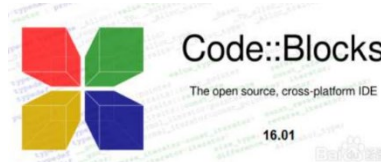
## 1.5 C++程序初探



# “Hello World” 程序

## 在屏幕上打印 “Hello World !”

在屏幕上打印  
hello world



集成开发环境(Integrated Development Environment)



构思

```
...  
cout<<"hello world";  
...
```

编写程序

```
11111111 11011000 11111111 11100000 00000000  
00010000 01001010 01000110 01001001 01000110  
00000000 00000001 00000010 00000000 00000000  
00000100 00000000 00000011 00000000 00000000  
11111111 11111110 00000000 00010000 01001100  
01100001 01110110 01100011 00110101 00110111  
00101110 00111000 00111001 00101110 00110001  
00110000 00110000 00000000 11111111 11011011  
00000000 01000011 00000000 00001000 00000100  
00000100 00000100 00000100 00000100 00000101  
00000101 00000101 00000101 00000101 00000101  
00000110 00000110 00000110 00000110 00000110  
00000110 00000110 00000110 00000110 00000110  
00000110 00000110 00000110 00000111 00000111
```

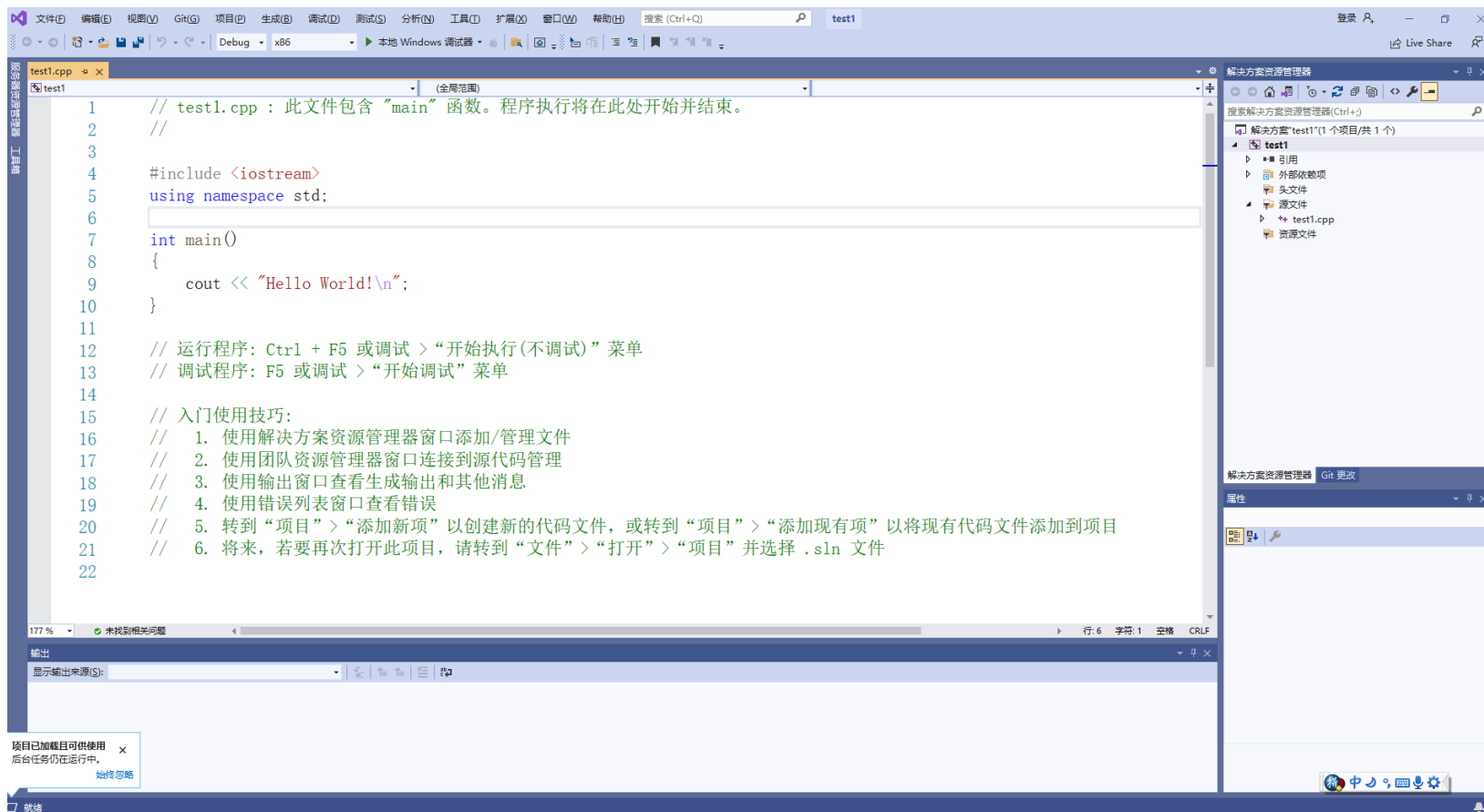
编译



执行

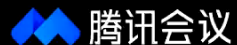
# 微软 Visual Studio

- ❑ 优点：不仅限于C++开发，比较全面
- ❑ 缺点：软件太庞大



# 编写并执行 “Hello World” 过程

---



## 李雨森的快速会议

会议号: 525 136 552

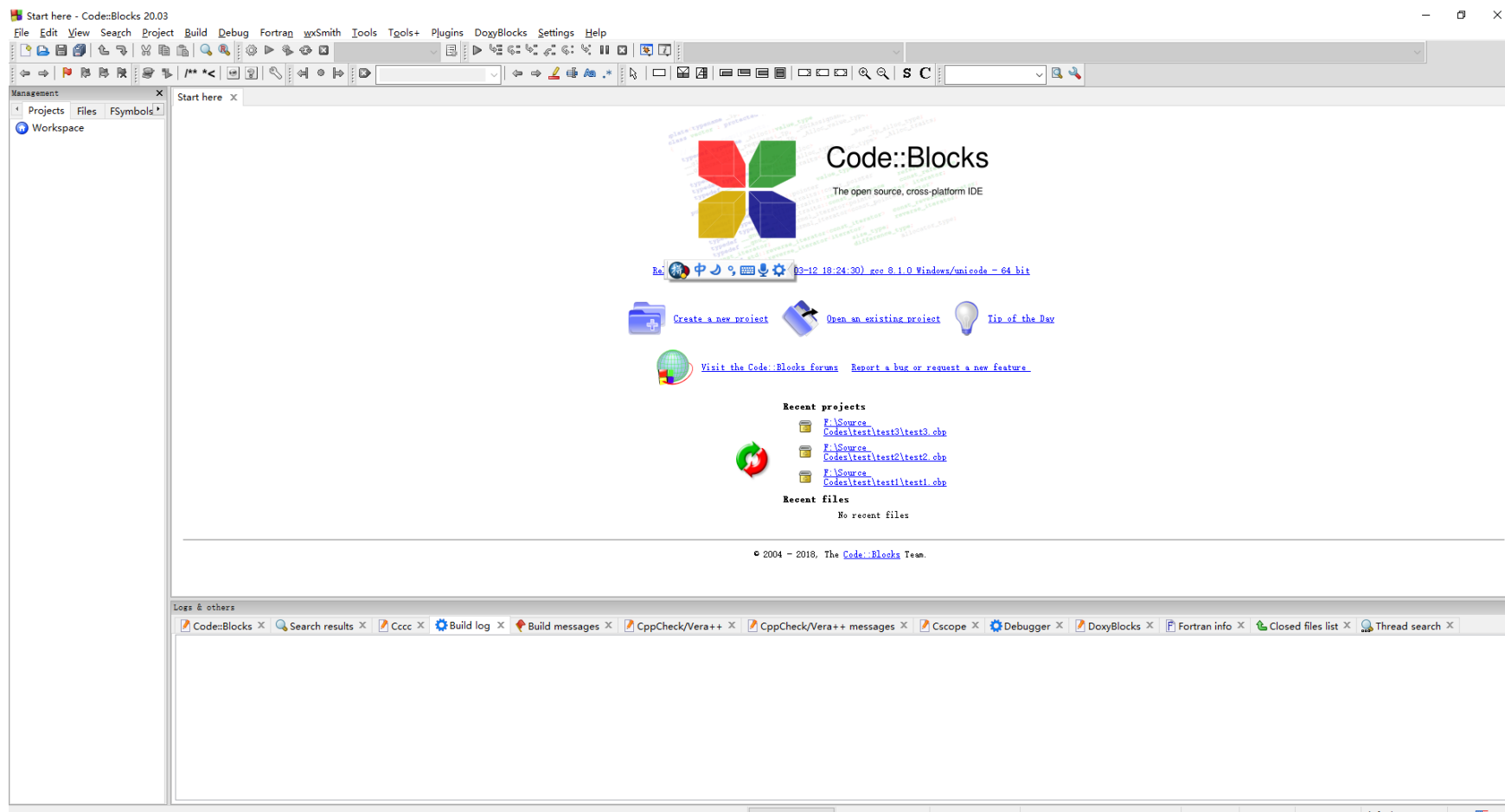
开始录制时间: 2021/09/08 16:35:01

创建者: 李雨森

# CodeBlocks

## 开源免费的IDE

□ 优点：轻量、高效



# CodeBlocks

---

CodeBlocks官方网站，直接下载安装即可：

<https://www.codeblocks.org/downloads/binaries/>

**NOTE:** The default builds are 64 bit (starting with release 20.03). We also provide 32bit builds for convenience.

---



## Microsoft Windows

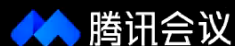
File	Download from
codeblocks-20.03-setup.exe	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>
codeblocks-20.03-setup-nonadmin.exe	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>
codeblocks-20.03-nosetup.zip	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>
codeblocks-20.03mingw-setup.exe	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>
codeblocks-20.03mingw-nosetup.zip	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>
codeblocks-20.03-32bit-setup.exe	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>
codeblocks-20.03-32bit-setup-nonadmin.exe	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>
codeblocks-20.03-32bit-nosetup.zip	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>
codeblocks-20.03mingw-32bit-setup.exe	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>
codeblocks-20.03mingw-32bit-nosetup.zip	<a href="#">FossHUB</a> or <a href="#">Sourceforge.net</a>

**NOTE:** The codeblocks-20.03-setup.exe file includes Code::Blocks with all plugins. The codeblocks-20.03-setup-nonadmin.exe file is provided for convenience to users that do not have administrator rights on their

---

# 编写并执行 “Hello World” 过程

---



## 李雨森的快速会议

会议号: 698 699 044

开始录制时间: 2021/09/08 19:01:45

创建者: 李雨森

# 在线平台

有很多在线平台,打开网页可直接编译、运行程序:

<https://www.bejson.com/runcode/cpp920/>

缺点: 可用性不能保证、不易调试

**BEJSON** 始于 2011

JSON工具 ▾ 编码/加密 ▾ 格式化 ▾ 网络 ▾ 前端 ▾ 后端 ▾ 转换 ▾ 其他 ▾ 文档 ▾ 平台工具 ▾ 更多 ▾

JSON生成Java实体类 | SQL生成Java代码 | JSON转C#实体类 | mysql转Java实体类 | SQL转CSharp实体类 | oracle转Java实体类 | 更多 ▾

什么是JSON | JSON的用法 | 腾讯云首单秒杀 | 香港服务器-新人专享3折起 | 阿里云新用户低至0.6折起 | 华纳云\_CN2 香港服务器18元/月

### C++在线运行(GCC 9.2.0)

```
hello.cpp +
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hello World";
6     return 0;
7 }
```

✕ C++ (GCC 9.2.0) ▾ 请输入命令行参数 运行(F9)

➡ Input Optional

➡ Output Empty



## 1.6 C++快速入门

所有C++程序必须用英文字符，不能用中文输入法输入

# 输出字符串

---

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

TIPS:

□ 字符串用双引号括起来

在控制台输出  
hello world!

\*程序的其余部分可认为是固定格式，先不用理解

---

# 输出字符串

---

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"nankai"<<endl;
    cout<<"university"<<endl;
    return 0;
}
```

TIPS:

- ❑ endl表示换行
- ❑ 如果程序有多行，每行结束后加分号；

在控制台输出  
nankai  
university

# 输出字符

```
#include<iostream>
using namespace std;
int main()
{
    cout<<'A' ;
    cout<<'B' <<endl;
    cout<<'C' <<' ' <<'2' ;
    return 0;
}
```

TIPS:

- ❑ 字符用单引号括起来
- ❑ 单引号中只能包含1个字符
- ❑ 一条cout语句可以连续多次输出

在控制台输出

AB

C 2

# 输出数值

---

```
#include<iostream>
using namespace std;
int main()
{
    cout<<2<<'\t'<<3.5<<endl;
    cout<<-4;
    return 0;
}
```

在控制台输出  
2 3.5  
-4

这里的数值2和前面例子中的字符'2'有何不同?

---

# 使用整数变量

---

```
#include<iostream>
using namespace std;
int main()
{
    int x = 3;
    int y = 5;
    cout<<x + y;
    return 0;
}
```

TIPS:

□ int是integer的缩写，表示整数类型

定义2个整数类型的变量 x, y

分别初始化为3和5，输出x+y的值

# 使用整数变量

```
#include<iostream>
using namespace std;
int main()
{
    int x, y;
    x = 3;
    y = 5;
    cout<<x + y;
    return 0;
}
```

TIPS:

- ❑ 变量可以先定义、后赋值
- ❑ 可以同时定义多个变量，用逗号隔开

定义2个整数类型的变量 x, y

分别初始化为3和5，输出x+y的值



# 使用整数变量

```
#include<iostream>
using namespace std;
int main() {
    int x = 3;
    int y = 5;
    cout<<x + y;
    x = 4;
    cout<<x + y;
    return 0;
}
```

TIPS:

□ 变量的值可以更改

定义2个整数类型的变量  
x, y

分别初始化为3和5，输出  
x+y的值

更改变量x的值，再次输出  
x+y的值

# 使用整数变量

```
#include<iostream>
using namespace std;
int main()
{
    int x = 3;
    int y = 5;
    int result = x - y;
    cout<<result;
    return 0;
}
```

定义2个整数类型的变量 x, y, 分别初始化为3和5

定义整数类型变量 result,初始化为x-y  
输出result

# 使用整数变量

---

```
#include<iostream>
using namespace std;
int main()
{
    int x = 3;
    int y = 5;
    int result = x - y;
    cout<<result;
    return 0;
}
```

## 变量的命名规则:

- 只能使用大小写字母、数字、下划线
- 首字符必须是字母或下划线
- 区分大小写字母
- 长度无限制，最好小于32
- 不能和系统关键字重名

# 使用浮点数变量

```
#include<iostream>
using namespace std;
int main()
{
    float x = 3.2;
    float y = 5.46;
    float result = x + y;
    cout<<result;
    return 0;
}
```

定义2个浮点数变量  
x, y, 分别初始化为  
3.2和5.46

定义浮点数变量  
result,初始化为x+y  
输出result

# 使用字符变量

---

```
#include<iostream>
using namespace std;
int main()
{
    char x = 'A' ;
    char y = 'B' ;
    cout<<x<<' '<<y;
    return 0;
}
```

定义2个字符型变量  
x, y, 分别初始化为  
'A'和'B'

输出x和y

# 输入整数

---

```
#include<iostream>
using namespace std;
int main()
{
    int x, y;
    cin>>x>>y;
    cout<<x + y;
    return 0;
}
```

TIPS: 程序执行时，从控制台输入两个整数（用空格隔开），按回车，看到输出结果

定义2个整数类型的变量  
 $x, y$

从键盘输入两个整数，  
分别赋值给 $x$ 和 $y$

输出 $x+y$

# 输入浮点数

---

```
#include<iostream>
using namespace std;
int main()
{
    float x, y;
    cin>>x>>y;
    cout<<x + y;
    return 0;
}
```

定义2个浮点型变量 x, y  
从键盘输入两个小数,  
分别赋值给x和y  
输出x+y

# 输入字符

```
#include<iostream>
using namespace std;
int main()
{
    char x, y;
    cin>>x>>y;
    cout<<x<<' '<<y;
    return 0;
}
```

TIPS:

□ 在键盘输入字符的时候  
不用加单引号

定义2个字符型变量 x, y  
从键盘输入两个字符,  
分别赋值给x和y  
输出x和y



# 输入字符串

```
#include<iostream>
using namespace std;
int main()
{
    string x;
    cin>>x;
    cout<<x;
    return 0;
}
```

TIPS:

- ❑ 在键盘输入字符串时以回车结尾
- ❑ 输入字符串无需加双引号
- ❑ 字符串中不能有空格

定义1个字符串类型的变量 x

从键盘输入一个字符串，赋值给x，输出x

# 程序注释

---

```
/*  
print "hello world" on the  
screen  
*/  
  
#include<iostream>  
using namespace std;  
  
int main()  
{  
    //cout<<"hello world! ";  
    return 0;  
}
```

## 注释

在程序上做的笔记，提高程序可读性，注释不会被编译和执行

注释有以下两种方式。

### (1) 单行注释

使用符号//，其后面直到一行结束的内容都为注释内容

### (2) 多行注释

使用符号/\*和\*/，其中间的内容为注释内容

END