

[1] 简简单单写程序

深入浅出程序设计竞赛
第 1 部分 – 语言入门
V 2021-04

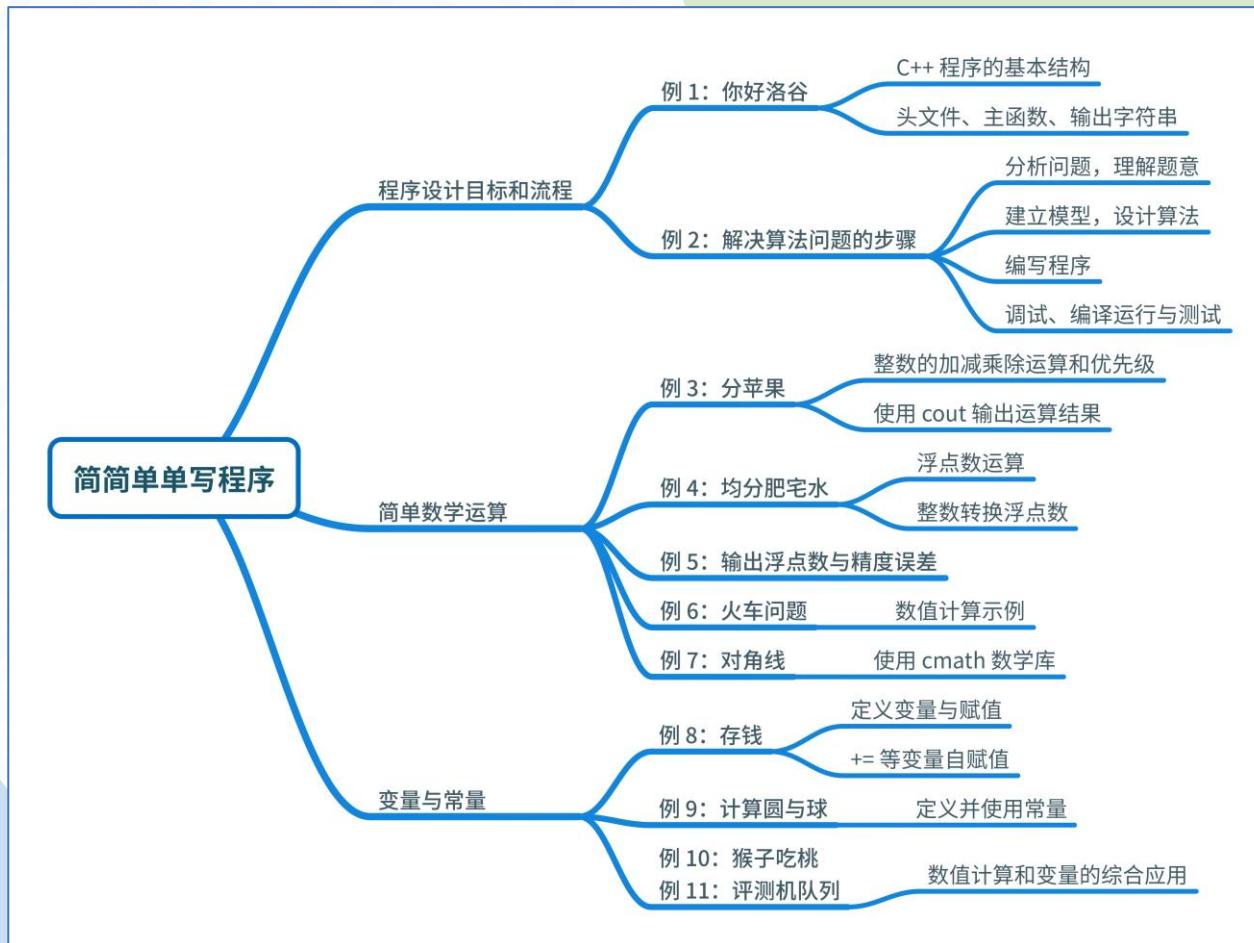
版权声明

本课件为《深入浅出程序设计竞赛 - 基础篇》的配套课件，版权归 洛谷 所有。所有个人或者机构均可免费使用本课件，亦可免费传播，但不可付费交易本系列课件。

若引用本课件的内容，或者进行二次创作，请标明本课件的出处。

- 其它《深基》配套资源、购买本书等请参阅：
<https://www.luogu.com.cn/blog/kkksc03/IPC-resources>
- 如果课件有任何错误，请在这里反馈
<https://www.luogu.com.cn/discuss/show/296741>

本章知识导图



第1章 简简单单写程序

程序设计的目标和流程

简单数学运算

变量与常量

课后习题与实验

程序的设计目标和流程

设计一个程序是为了让计算机始终不渝地遵循指令，以完成特定的任务。为了能让计算机听懂指令，我们编写程序来与计算机交流。

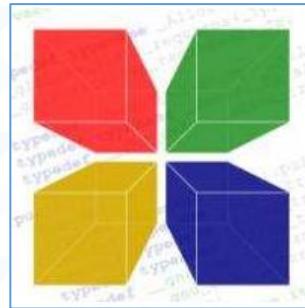
请翻至课本 P2

编程方法

使用IDE（集成编辑环境）

参阅【附录A 程序设计程序配置】

例如：Dev C++ / CodeBlocks



使用洛谷在线编程

<https://www.luogu.com.cn/ide>

A screenshot of the Luogu online IDE interface. The top bar shows "C++11". The code editor contains the following C++ code:

```
</> 代码
1 #include<iostream>
2 #include<cmath>
3 // 填上你觉得需要的其他头文件
4 const double PI = 3.141593;
5 using namespace std;
6 int main() {
7     int T;
8     cin >> T;
9     if (T == 1) {
10         // 粘贴问题 1 的主函数代码，除了 return 0
11         cout << "I love Luogu!";
12     } else if (T == 2){
13         // 粘贴问题 2 的主函数代码，除了 return 0
14         cout << 2 + 4 << " " << 10 - 2 - 4;
15     } else if (T == 3){
16         cout << 14 / 4 << endl;
17         cout << 14 / 4 * 4 << endl;
18         cout << 14 - 14 / 4 * 4 << endl;
19     } else if (T == 4){
20         cout << 500.0 / 3 << endl;
21 }
```

The input and output panels are both empty. A "运行" (Run) button is located at the bottom right of the interface.

什么是编程

程序 就是为实现特定目标或解决特定问题而用计算机语言编写的一系列令序列。

计算机语言 是人与计算机之间通信的语言。

编程 是程序员为解决特定问题，按照自己的思路，在遵循特定的计算机语言规则下编写程序的过程。

C++ 语言

本课程以 C++ 语言规则 为例来学习编程。

直接原因 信息学竞赛只允许使用 C++

其他原因 比较底层，运行速度快

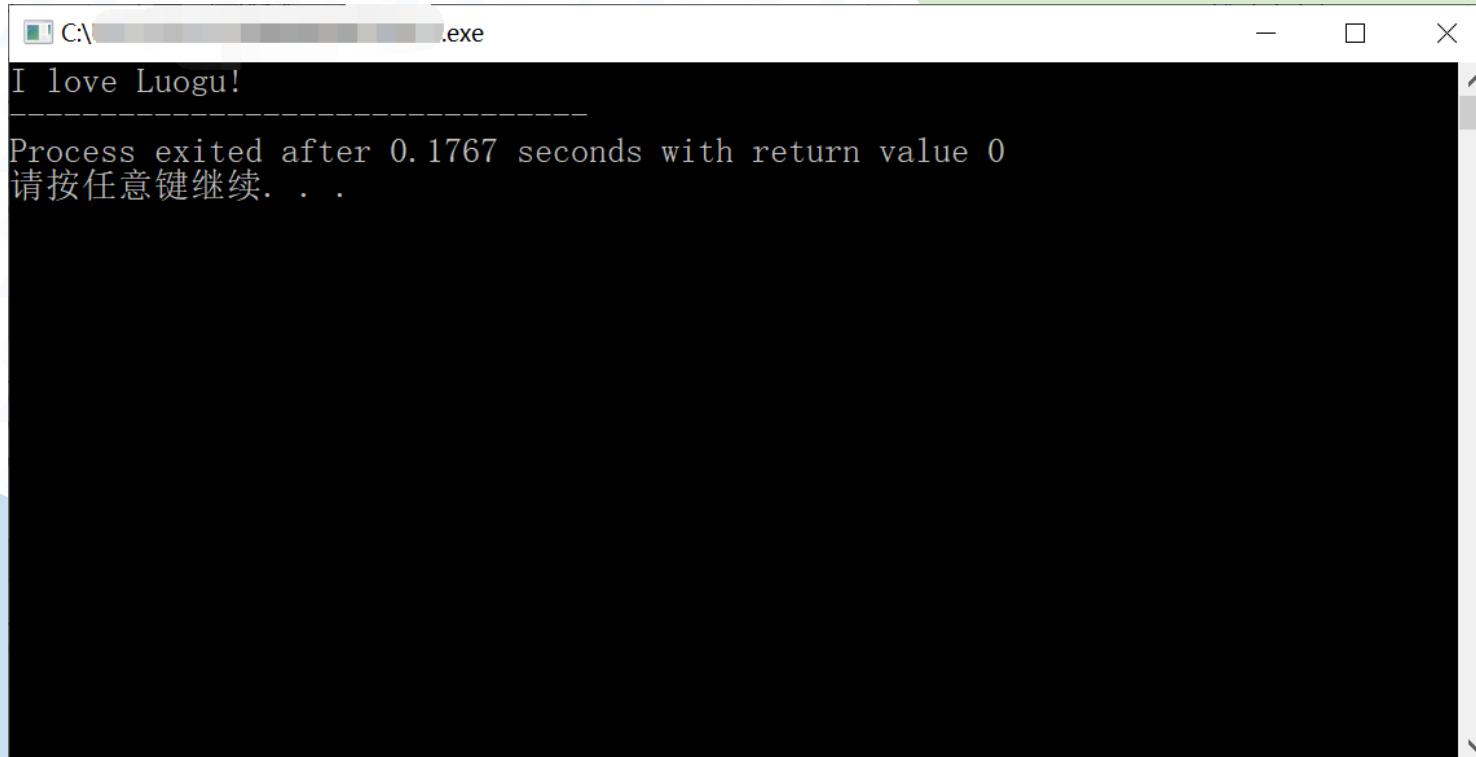
C++ 是编译语言，需要编译成可执行文件
(Windows 下是 exe)



你好洛谷

例 1.1

编写一个程序，让计算机告诉大家我爱洛谷，即 I love Luogu!



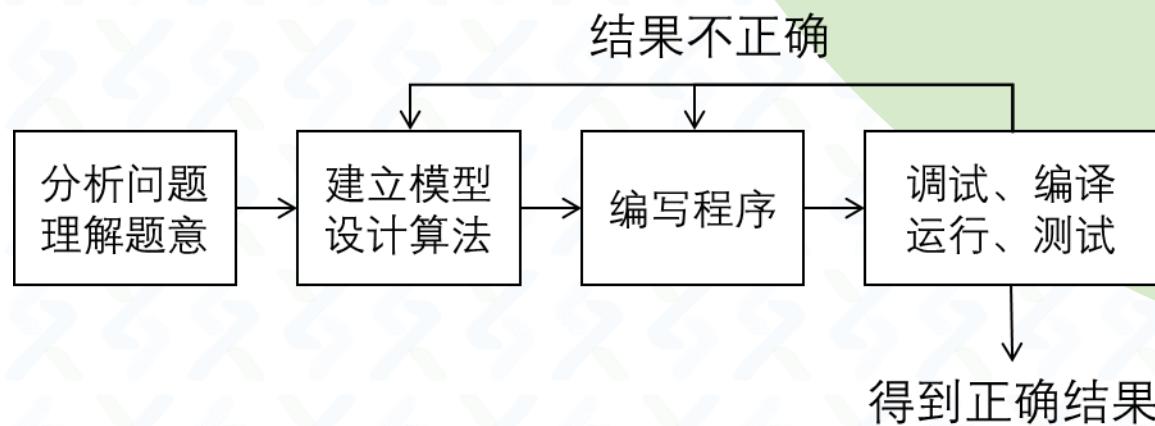
A screenshot of a terminal window titled "C:\ [REDACTED].exe". The window contains the following text:
I love Luogu!
Process exited after 0.1767 seconds with return value 0
请按任意键继续. . .

你好洛谷

```
#include<iostream>           → 头文件 引入外部库  
using namespace std;          → 命名空间 形成相对隔离空间  
  
int main() {                  → 主函数 程序的入口  
    cout << "I love Luogu!";   → 输出语句 输出一些内容  
    return 0;                   → 返回值 退出函数并返回值  
}
```

如果想借助计算机解决实际问题，就要设计计算机程序并让计算机执行。

解决算法问题的步骤



分苹果

例 1.2

这里有 10 个苹果，小A 拿走了 2 个，Uim 拿走了 4 个，八尾勇拿走剩下的所有的苹果。我们想知道：

- 小A 和 Uim 两个人一共拿走多少苹果？
- 八尾勇能拿走多少苹果？

现在需要编写一个程序，输出两个数字作为答案，中间使用空格分开。

分苹果

cout 语句分别输出了 $2+4$ 、`" "`、 $10-2-4$ 这三个项目。

这说明 cout 语句可以同时输出多个项目，中间用 `<<` 分割。

```
#include<iostream>
using namespace std;
int main() {
    cout << 2+4 << " " << 10-2-4;
    return 0;
}
```

运算符 `+` 是 加法

运算符 `-` 是 减法

输出多项数学运算

每项之间用 `<<` 隔开

不要忘记最后的 分号

简单数学运算

计算机，顾名思义，就是非常擅长做“计算”的机器。

在上一节中已经介绍了计算加减的方式，现在会告诉大家更多的计算符号。

请翻至课本 P6

分苹果 (2)

例 1.3

现在有 14 个苹果。要均分给 4 名同学，分不掉的苹果放回冰箱。
请问：

- 每位同学能分得几个苹果？
- 一共分出去多少苹果？
- 把几个苹果放回冰箱？

现在需要编写一个程序，输出三个数字作为答案，**每个数字一行**。

分苹果 (2)

```
#include<iostream>
using namespace std;
int main() {
    cout << 14 / 4 << endl;
    cout << 14 / 4 * 4 << endl;
    cout << 14 - 14/4*4 << endl;
    // cout << 14 % 4 << endl;
    return 0;
}
```

运算符 `/` 是 整除 (去掉余数后取整)

运算符 `*` 是 乘法

运算符 `%` 是 取余 (除不尽的部分)

$14/4*4$ 可不是 14 哦

算法优先级 括号最先, 然后乘除,
然后加减

`//` 开始到末尾结束的语句, 是 注
释, 程序会忽略

浮点数运算

浮点数运算，可以处理小数

注意 500.0 不是 500

前者是 浮点数，后者是 整数

运算符 / 是 除法（如果前后有至少一项浮点数的话）

注意：如果 / 的前后都是整数，那么就是整除。

加减乘除前后 有一项是浮点数，则结果为 浮点数

均分肥宅水

例 1.4

现在有 500 毫升的肥宅快乐水，要均分给 3 名同学，每位同学可以分到多少毫升？

请输出一个数字作为输出。

整数：500

浮点数：500.0 或 500*1.0

浮点数可以带小数点
但精度和范围有限。

```
#include<iostream>
using namespace std;
int main() {
    cout << 500.0 / 3 << endl;
    return 0;
}
```

上机实验

例 1.5

输入以下程序，观察输出。并且解释这样输出的原因

如何用科学计数法来表述实数呢？

```
#include<iostream>
using namespace std;
int main() {
    cout << 500.0 / 3 << endl;
    cout << 5000000.0 / 3 << endl;
    cout << 0.000005 / 3 << endl;
    cout << 5e6 / 3 + 5e-6 / 3 - 5e6 / 3 << endl;
    return 0;
}
```

166.667
1.66667e+06
1.66667e-06
1.6666e-06

火车问题

例 1.6

甲列火车长 260 米，每秒行 12 米；

乙列火车长 220 米，每秒行 20 米。

两车相向而行，从两车车头相遇时开始计时，多长时间后两车车尾相离？已知答案是整数。

提示： $(260 + 220) / (12 + 20)$



火车问题

提示：假设你坐在甲车上看乙车，乙车相对你的速度是 $12+20$ 。
乙车从甲车头部碰到，直到尾部离开，路程 $260+220$ 。

最后答案： $(260 + 220) / (12 + 20)$



```
#include<iostream>
using namespace std;
int main() {
    cout << (260 + 220) / (12 + 20) << endl;
    return 0;
}
```

由于已经保证题目是整数，所以可以直接使用整除。
但如果不能保证，则需要类型转换，例如整数后面加`.0`。

对角线

例 1.7

一个长方形长宽分别是 6cm、9cm，求它的对角线长度。

$$6^2 + 9^2 = ?^2$$

对角线

新的头文件 <cmath> 数学工具库

pow(a,b) 计算 a 的 b 次方，返回浮点数

sqrt(a) 计算括号里的算术平方根，返回浮点数

以上被称为 函数

```
#include<iostream>
#include<cmath>
using namespace std;
int main() {
    cout << sqrt(pow(6, 2) + pow(9, 2)) << endl;
    return 0;
}
```

对角线

cmath 头文件常用函数，暂时不要求全部记忆

函数原型	样例	说明
double sin(double x) double cos(double x)	sin(3.14159/2)	三角函数正弦和余弦，x 是弧度角度
double exp(double x)	exp(1)	返回 e^x ，其中 e 是自然对数的底
double log(double x)	log(10)	返回 x 的自然对数
double pow(double x, double y)	pow(3, 2)	返回 x^y 。也可以用来运算多次根式。
double sqrt(double x)	sqrt(9)	返回 \sqrt{x}
double fabs(double x)	fabs(-10)	返回 x 的绝对值
double ceil(double x)	ceil(2.1)	返回大于或等于 x 的最小的整数值（上取整）
double floor(double x)	floor(2.9)	返回小于或等于 x 的最大的整数值（下取整）

变量与常量

有的问题是需要通过多个表达式计算才能解决的，因此需要有一个办法能够储存计算中间结果。这时，可以使用变量来满足这个要求。

请翻至课本 P9

变量

用于 **存放数据** 的容器。

有不同的类型——整数、浮点数、字符，等等。

变量的值 **可以变化**。

可以当做表达式的一部分参与计算



int

变量有很多种，int 是整数类型

占用4字节的内存

可以存下 -21亿 到 21亿

定义一个 int 类型变量，名字是 a，其初始值为 1。

```
int a = 1;
```

存钱

例 1.8

Uim 银行账户里面有100元。经过了下面的操作：

1. 往里面存了 10 元；
2. 购物花掉了 20 元；
3. 把里面的钱全部取出。

请在每次操作后输出账户余额，并使用换行符隔开。

存钱

```
#include<iostream>
using namespace std;
int main() {
    int balance = 100; // 初始余额
    balance = balance + 10;
    cout << balance << endl;
    balance -= 20;
    cout << balance << endl;
    balance = 0;
    cout << balance << endl;
    return 0;
}
```

定义了 balance 这个 int (整数)类型的变量，初始化为 10

balance = balance + 10 是赋值语句

balance -= 20 是自赋值语句

自赋值还能 +=、 *=、 /=、 %=

变量赋值一个数字或者表达式

常量

常量，跟变量差不多，但是加上 `const`

- 例如 `const int N=10` 代表定义一个类型为 `int`，名称为 `N` 的常量，其值 恒定为 10。`const` 和 `int` 的顺序可以交换或者使用 宏定义，对代码片段进行粗暴的替换
- 例如 `#define N 10` 代表将元素 `N` 全部替换为 10

常量不能变化，习惯上用大写字母代替

计算圆和球

例 1.9

当半径为 $r = 5$, 请输出圆的周长、面积和球体积。取 $\pi = 3.141593$

圆的周长是 $C = 2\pi r$, 面积是 $S = \pi r^2$, 球体积是 $V = \frac{4}{3}\pi r^3$

计算圆和球

```
#include<iostream>
#include<cmath>
using namespace std;
int main() {
    double r = 5;
    const double PI = 3.141593;
    // #define PI 3.141593
    cout << 2 * PI * r << endl;
    cout << PI * r * r << endl;
    cout << 4.0/3*PI*pow(r,3)<<endl;
    // 不能写成 4/3*PI*pow(r,3)
    return 0;
}
```

PI 被定义为 常量，跟变量差不多，但是加上 const double 是代表浮点数
这里 PI 被固定为 3.141593，
不能被修改

下面使用到 PI 时，都会引用这个数
或者使用 #define 宏

变量的命名

1. 只能由 英文字母、数字和下划线（_） 组成；
2. 不能以数字开头；
3. 不能和其他“关键字”重复。关键字（又称保留字）有很多，比如int、if 等等。
4. 区分大小写

判断练习

以下变量名是否合法？

1. kkksc03
2. OhILoveLuoguVeryMuchAndIWillStudy
3. _1apple
4. char
5. kkk@SH
6. a
7. iPhone
8. 11dimensions
9. __stdcall (两个下划线)

判断练习

答案

1. kkksc03 (合法)
2. OhILoveLuoguVeryMuchAndIWillStudy (合法)
3. _1apple (合法)
4. char (不合法, 类型保留字)
5. kkk@SH (不合法, 有@)
6. a (合法)
7. iPhone (合法)
8. 11dimensions (不合法, 数字开头)
9. __stdcall (看情况)

猴子吃桃

例 1.10 (洛谷P5743)

一只小猴买了若干个桃子。

第一天他刚好吃了这些桃子的一半，又贪嘴多吃了1个；

第二天他也刚好吃了剩余桃子的一半，贪嘴多吃了1个；

第三天他又刚好吃了剩下的桃子的一半，并贪嘴多吃了1个。

第四天起来一看，发现桃子只剩下一个了。

请问小猴买了几个桃子？

猴子吃桃

```
#include<iostream>
using namespace std;
int main() {
    int num = 1; // 第四天
    num = (num + 1) * 2; // 第三天
    num = (num + 1) * 2; // 第二天
    num = (num + 1) * 2; // 第一天
    cout << num << endl;
    return 0;
}
```

从第四天逆着推回来！

定义变量，存储中间结果

注意 = 是赋值符号，不是代表相等

$((1+1)*2+1)*2+1)*2$ 这种写法
正确但是不直观

课后习题与实验

学而时习之，不亦说乎。学而不思则罔，思而不学则殆。——孔子

请翻至课本 P12

填空练习

请将下列公式翻译成表达式

$$3x + 5y$$

$$\frac{c + 1}{ab}$$

$$\sqrt{3a^3}$$

$$(n + 2)(n - 9)$$

填空练习

答案

$$3x + 5y$$

`3*x+5*y`

$$\frac{c + 1}{ab}$$

`(c+1)/(a*b)` 或者 `(c+1)/a/b`

$$\sqrt{3a^3}$$

`sqrt(3*a*a*a)` 或者 `sqrt(3*pow(a,3))`

$$(n + 2)(n - 9)$$

`(n+2)*(n-9)`

复习

程序设计流程 分析问题 建立模型 编写程序 调试与测试
程序基本框架 头文件 名字空间 主函数 输出语句 返回值
加减乘除的运算 运算优先级

整数与浮点数

使用数学函数 <cmath>

变量: 用于存放数据的容器。有不同的类型，值可以变化。可以当做表达式的一部分参与计算。变量的命名有讲究

常量: 用于表达固定的常数。

可以使用变量分步骤完成比较复杂的计算。

作业

1. 参考《深基》附录，将本课例题代码在本地运行
2. 习题 1.4

编程解决应用问题，然后手工验算，验证答案
一共有7个小问。

题目详见课本 P13



超级玛丽游戏

编程练习 – 洛谷 P1000

超级玛丽是一个非常经典的游戏。请你用字符画的形式输出超级玛丽中的一个场景。

提示：

```
cout<<"      ****      "<<endl;
```