

内容回顾

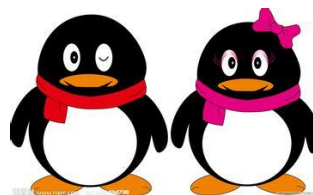
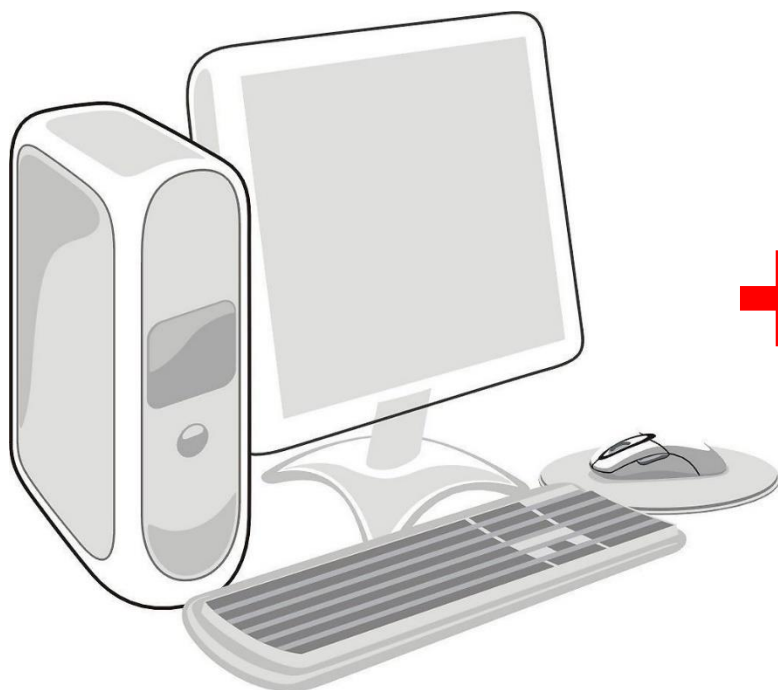
莫希利
埃克特 (24岁)、
宾夕法尼亚
大学

冯·诺依曼
(John von
Neumann),
1903 – 1957

ENIAC(Electronic Numerical Integrator And Calculator):

世界上第一台电子多用途计算机 1946年2月14日 (1942年提出研制设想)

内容回顾



....

硬件

计算机物理构成
是“物质基础”
决定计算性能！

软件

计算机程序
像“灵魂”
定义功能！

内容回顾

教师是人类灵魂的“工程师”！

如何做计算机灵魂的工程师？

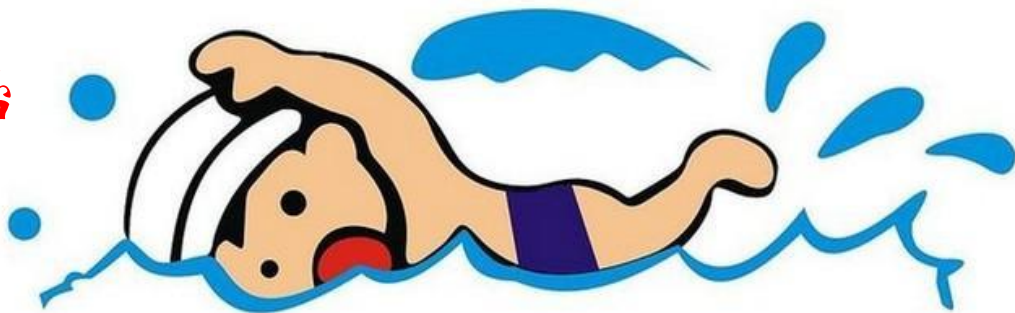
从学习程序设计基础开始吧！



内容讲述顺序 V. S. 课程章节设置

- ▶ 概述与初识C/C++程序（教材第1章）
- ▶ 程序的流程控制（教材第3章）
- ▶ 过程抽象与封装—函数（教材第4章）
- ▶ 简单数据的描述与操作（教材第2章）
- ▶ 构造数据类型（教材第5章）
- ▶ 面向对象的程序设计（教材第6章）

让大家更早的开始
一试身手！
写程序！



关于课程学习 - 本学期最重要的课程!

- ▶ 扎实掌握基础知识 + 注重动手能力
- ▶ 对初学者，关键在“**练**”：**编程、编程、再编程、调试、调试、再调试...**
 - ▶ 完成但不要局限于书本习题!
 - ▶ 充分利用上机时间...
 - ▶ 周二下午-晚上7-10节 基础实验楼乙125
 - ▶ 有限的课程上机时间还是不够的...
- ▶ 利用网络...
- ▶ 严格遵守课堂纪律!
 - ▶ 认真听讲、“不用做笔记”
- ▶ 严谨玩手机、接电话、刷微信...

课程目标!!!

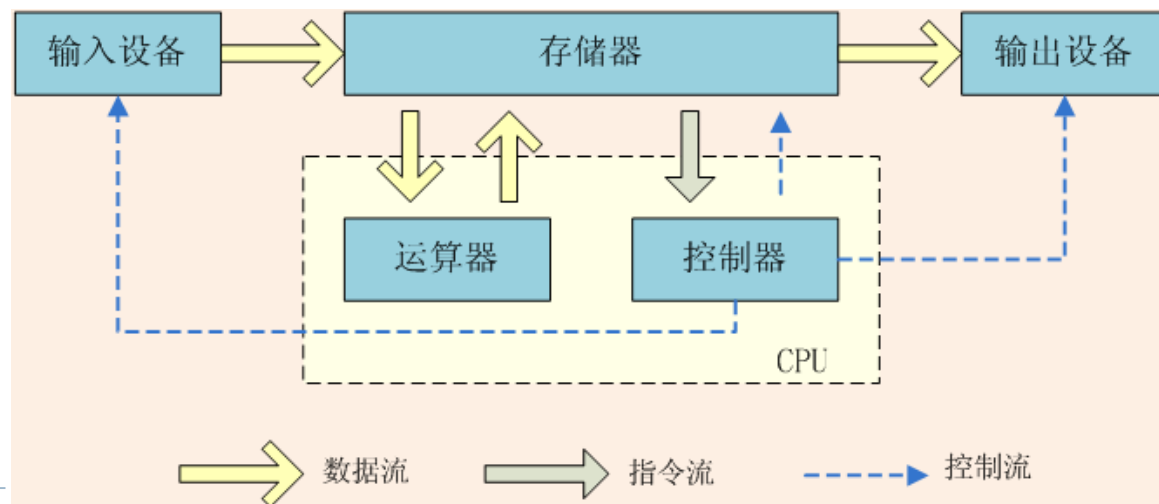
- ▶ 掌握基本的程序设计概念和方法
 - ▶ 熟练掌握 **C/C++数据类型、语法结构等**
 - ▶ 掌握 **基本**的排序等 **算法**，能够熟练 **编写程序**
 - ▶ 熟练掌握程序的 **调试方法！保证程序高效、正确运行！解决一般问题**
 - ▶ 培养良好的编程习惯！
 - ▶ 为后续课程打基础
 - ▶ 程序设计实验（大一下）、高级程序设计（大二）?、数据结构（大二）、算法、...
-

冯·诺依曼体系结构

- ▶ 逻辑上，冯·诺依曼计算机由5个单元构成
 - ▶ **输入**单元：从外界获得数据
 - ▶ **存储**单元：存储程序（指令序列）和数据
 - ▶ **运算**单元：进行算术/逻辑运算
 - ▶ **控制**单元：控制程序的执行和根据指令向其它单元发出控制信号
 - ▶ **输出**单元：向外界输出结果

冯·诺依曼计算机工作过程

- ▶ 借助**输入单元**把待执行的程序装入到存储单元；
- ▶ **控制单元**从**存储单元**中逐条地读取程序中的指令执行，程序执行中：从**输入单元**或**存储单元**中获得所需要的数据，把其中的计算指令交给**运算单元**完成；
- ▶ 程序执行产生的临时结果保存在存储单元中，程序的最终执行结果通过**输出单元**输出



关于上机和平时学习的问题

- ▶ 关于上机：

- ▶ 基础实验楼乙125（坐固定位置！）

- ▶ 关于答疑：

- ▶ 线下：我（计算机楼1004，提前预约） +
助教（计算机楼314，晚上）

- ▶ 鼓励自己探索！



第一次上机:

从一个完整的程序例子开始!

- ▶ 例0 计算一组圆（直径为n以内的正整数）的周长之和（计量单位为米）。

```
#include <stdio.h>
```

```
#include <stdio.h>
```

```
#define PI 3.14
```

```
int main( )
```

```
{ int n, d = 1;  
  double sum = 0;  
  char ch = 'm';  
  printf("Input n: ");  
  scanf("%d", &n);
```

```
.....
```

```
return 0;
```

```
}
```

```
while(d <= n)
```

```
{
```

```
    sum = sum + PI * d;
```

```
    d = d + 1;
```

```
}
```

```
printf("The sum is: %f ", sum);
```

```
putchar(ch); //显示计量单位
```

2 初识C/C++程序（上）

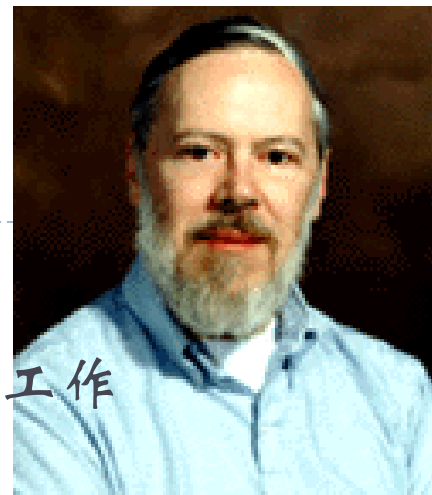
郭延文

2019级 计算机科学与技术系

C/C++语言的来历

- ↓
 - ▶ ALGOL 60 (algorithmic language, 国际委员会, 1960)
- ↓
 - ▶ C (combined programming language, 剑桥、伦敦大学, 1963)
- ↓
 - ▶ BCPL (basic ~, 剑桥大学Martin Richards, 1967)
- ↓
 - ▶ B (贝尔实验室Ken Thompson, 1970)
- ↓
 - ▶ C (贝尔实验室D.M.Ritchie, 1972~1973)
 - ▶ 既保持了BCPL和B语言的优点(精练、高效、接近硬件等)
 - ▶ 又克服了它们的缺点(过于简单、数据无类型、功能有限等)
- ↓
 - ▶ C++ (贝尔实验室Bjarne Stroustrup, 1979)
 - ▶ 为支持面向对象程序设计而设计(先是C with Class)

C语言之父



▶ Dennis M. Ritchie(1941-2011)

- ▶ 1967起一直在位于美国新泽西州的贝尔实验室工作
- ▶ 他的工作得到了很多计算机组织的公认和表彰
 - ▶ 美国计算机协会(**ACM**)授予的系统及语言杰出论文奖(1974)
 - ▶ 电气和电子工程师协会(**IEEE**)的先驱奖(Emmanuel Piore)(1982)
 - ▶ **图灵奖**(Turing)(1983)
 - ▶ ...

Association for **C**omputing **M**achinery

Institute of **E**lectrical and **E**lectronics **E**ngineers

C/C++语言标准

- ▶ 1978年Dennis M. Ritchie与Brian W.Kernighian出版了《The C Programming Language》，此书已译成多种语言，成为C语言方面最权威的教材之一，是最初的C语言标准（K&R C）
- ▶ 随着C语言的应用和发展，形成了多种C语言版本，各种版本在功能和函数库的设置内容上存在差别。1983年，ANSI开始制定统一的C语言标准，直至1989年底正式批准名为ANSI X3.159-1989的标准（C89），1990年，ISO采纳了C89并以ISO/IEC 9899:1990颁布

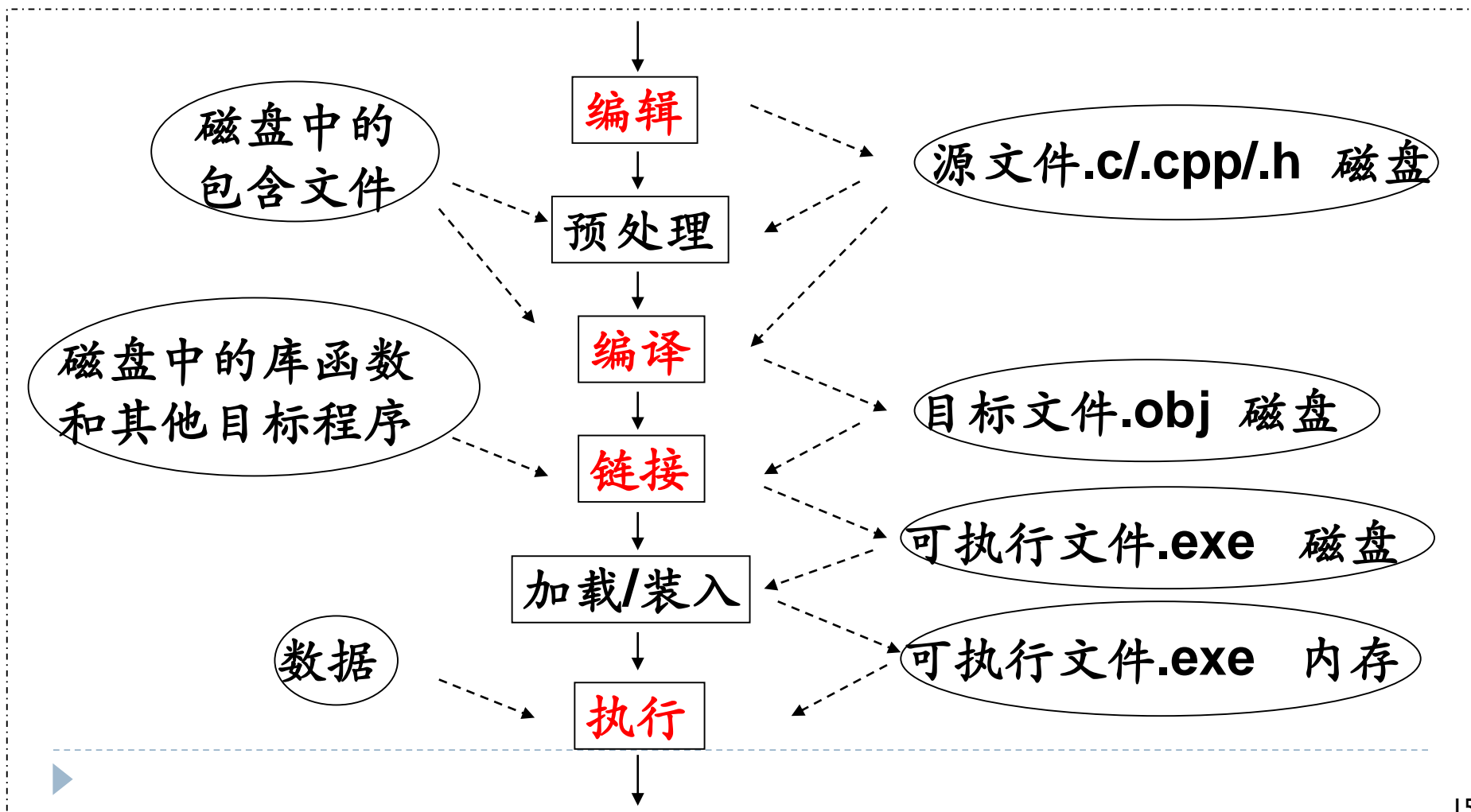
American National Standards Institute

International Organization for Standardization

- ▶ 2000年初，ISO颁布了ISO/IEC 9899:1999（C99）
- ▶ 2011年底，ISO发布了ISO/IEC 9899:2011（C11）
- ▶ 1998年底，ISO发布了ISO/IEC 14882:1998（C++98）
- ▶ 2011年底，ISO发布了ISO/IEC 14882:2011（C++11）

C/C++程序的运行步骤

► 编辑(.c/.cpp/.h) → 编译(.obj) → 链接(.exe) → 执行 → 输出结果



C/C++程序开发环境

- ▶ 上述C/C++程序的编辑、编译等步骤都需要特定软件的支撑。
- ▶ **调试(Debug):** 操作过程中, 有可能发现程序有错, 需要修改程序, 整个过程可能会重复多次, 直到得出正确的执行结果。
- ▶ 通常使用带有编辑器、编译器等支撑软件的集成开发环境 (IDE) 进行程序开发。
 - ▶ **Windows:** Visual Studio 或 Eclipse
 - ▶ **Linux:** gcc+gdb, vim+gcc
 - ▶ **通用平台:** QT, codeblock
- ▶ 在IDE中, 往往使用一条命令 (菜单) 就能完成所有步骤。一些IDE提供可视化的程序设计支持和功能强大的程序动态调试 (Debug) 等工具。

C/C++程序的基本结构与main函数

- ▶ 一个简单的C/C++程序只有一个名为main的函数（子程序）
- ▶ 一个复杂的C/C++程序可以有多个函数，但其中必须有且只有一个main函数（不妨设其位于文件main.c/main.cpp中），


这些函数可以位于一个文件中，也可以位于多个源文件中

```
int main(void)
{
    //.....
    return 0;
}
```

- 程序运行时，从main函数中的第一条语句开始执行，执行到main函数中的return语句结束。
- return后面的值返回给操作系统，以便让操作系统了解程序运行的情况，一般地，0表示程序正常结束，-1表示程序异常结束。
- main前面的int（整型）是main函数的类型，与return后面的0相一致。
- 一般情况下，main函数不带参数，所以参数类型可以写成void（空类型），void可以省略。

关于main函数的返回值

```
#include <stdio.h>
int main()
{
    printf("Hello world!\n");
    return 0;
}
```



```
#include <stdio.h>
main()
{
    printf("Hello world!\n");
}
```

```
#include <stdio.h>
void main()
{
    printf("Hello world! \n");
}
```

关于void main()

- ▶ VS环境下编译通过!
- ▶ 不合标准 // C/C++标准中是int main()
- ▶ 不能保证正常工作//可能是微软发明的
- ▶ 不具可移植性 //程序在不同平台移植可能出现问题

一个完整的例子

- ▶ 例0 计算一组圆（直径为n以内的正整数）的周长之和（计量单位为米）。

```
#include <stdio.h>
```

```
#define PI 3.14
```

```
int main( )
```

```
{ int n, d = 1;  
  double sum = 0;  
  char ch = 'm';  
  printf("Input n: ");  
  scanf("%d", &n);
```

```
.....
```

```
return 0;
```

```
}
```

```
while(d <= n)
```

```
{
```

```
    sum = sum + PI * d;
```

```
    d = d + 1;
```

```
}
```

```
printf("The sum is: %f ", sum);
```

```
putchar(ch); //显示计量单位
```

一个完整的例子 - 逐句解释

- ▶ 例0 计算一组圆（直径为n以内的正整数）的周长之和（计量单位为米）。

```
#include <stdio.h>           // 编译预处理命令 标准输入、输出，
                                // 如果要用printf/scanf 必须有这句
#define PI 3.14               // 下面用PI表示3.14
int main( )                   // C语言主函数，程序唯一入口
{
    int n, d = 1;              // 定义整形变量n和d (表示直径，初始值是1)
    double sum = 0;            // 定义浮点型sum (表周长和，初始值是0)
    char ch = 'm';             // 定义字符型ch (值为m)
    printf("Input n: ");       // 在console (一个黑窗口) 下打印出 Input n:
    scanf("%d", &n);           // 让用户在键盘下输入n的值，整数

    .....
    return 0;                  // 返回值0，表示程序正常结束
}
```

一个完整的例子 - 逐句解释

- ▶ 例0 计算一组圆（直径为n以内的正整数）的周长之和（计量单位为米）。

```
while(d <= n)           // while循环语句，循环n次，直到d>n
{
    sum = sum + PI * d; // 赋值，用sum + PI * d更新sum
    d = d + 1;          // 赋值，用d+1更新d
}

printf("The sum is: %f ", sum); putchar(ch);
// 在黑窗口下打印出, The sum is ... ; 显示计量单位
```

一个完整的例子（C++兼容的C）

- ▶ 例0 计算一组圆（直径为n以内的正整数）的周长之和（计量单位为米）。

```
#include <stdio.h>
```

```
#include <stdio.h>
```

```
#define PI 3.14
```

```
int main( )
```

```
{ int n, d = 1;  
  double sum = 0;  
  char ch = 'm';  
  printf("Input n: ");  
  scanf("%d", &n);
```

```
.....
```

```
return 0;
```

```
}
```

```
while(d <= n)
```

```
{
```

```
    sum = sum + PI * d;
```

```
    d = d + 1;
```

```
}
```

```
printf("The sum is: %f ", sum);  
putchar(ch); //显示计量单位
```

解读C/C++的“=”

```
while(d <= n)
{
    sum = sum + PI * d;
    d = d + 1;
}

printf("The sum is: %f ", sum);
putchar(ch); //显示计量单位
```

► 赋值符号

// 第一次循环
开始d是1;
经过一次循环,
sum由0变为PI
d由1变为2;

// 第二次循环
d开始是2;
经过一次循环,
sum由PI变为3*PI
d由2变为3;

解读 scanf函数; // scanf("%d", &n);

scanf是C语言中的标准输入函数

函数原型如下:

```
int scanf( const char *format, ... );
```

其调用格式为:scanf("<格式化字符串>", <地址表>);

举例:

```
int a;  
char b;  
float c;  
char d[10];  
scanf("%d",&a);  
scanf("%c",&b);  
scanf("%f",&c);  
scanf("%s",&d);
```



一个完整的例子 (C++)

- ▶ 例0 计算一组圆（直径为n以内的正整数）的周长之和（计量单位为米）。

```
#include <stdio.h>
```

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

```
#define PI 3.14
```

```
const double PI = 3.14;
```

```
int main( )
```

```
{ int n, d = 1;
```

```
double sum = 0;
```

```
char ch = 'm';
```

```
printf("Input n: ");
```

```
scanf("%d", &n);
```

```
.....
```

```
return 0;
```

```
}
```

```
while(d <= n)
{
```

```
sum = sum + PI * d;
d = d + 1;
```

```
cout << "Input n: " ;
cin >> n;
```

```
printf("The sum is: %f ", sum);
putchar(ch); //显示计量单位
```

```
cout << "The sum is: " << sum;
cout << ch;
```

再回首：一个完整的例子（C）

- ▶ 例0 计算一组圆（直径为n以内的正整数）的周长之和（计量单位为米）。

```
#include "stdio.h"
```

```
#define PI 3.14
```

```
int main( )
```

```
{ int n, d = 1;  
  double sum = 0;  
  char ch = 'm';  
  printf("Input n: ");  
  scanf("%d", &n);
```

```
.....
```

```
  return 0;
```

```
}
```

```
while(d <= n)
```

```
{
```

```
    sum = sum + PI * d;
```

```
    d = d + 1;
```

```
}
```

```
printf("The sum is: %f ", sum);  
putchar(ch); //显示计量单位
```

再回首：一个完整的例子（C++）

- ▶ 例0 计算一组圆（直径为n以内的正整数）的周长之和（计量单位为米）。

```
#include <iostream>
using namespace std;
const double PI = 3.14;
```

```
int main( )
{ int n, d = 1;          //
  double sum = 0;        //
  char ch = 'm';
  cout << "Input n: " ;
  cin >> n;

  .....
  return 0;
}
```

```
while(d <= n)
{
    sum = sum + PI * d;
    d = d + 1;
}
```

```
cout << "The sum is: " << sum;
cout << ch;
```

C/C++输入输出用法对比

```
#include <stdio.h>
```

输入: `scanf("%d", &n);`

输出: `printf("Input n: ");`

- `include` 称为文件包含命令, 其意义是把尖括号"`<>`"或引号"`<>`"内指定的文件包含到本程序中, 成为本程序的一部分。被包含的文件通常是由系统提供的, 其扩展名为`.h`
- 而`stdio`为`standard input output`的缩写, 意为“标准输入输出”

v.s.

```
#include <iostream>
```

```
using namespace std;
```

输入: `cin >>`

输出: `cout <<`

- 标准C++引入了名字空间的概念, 并把`iostream`等标准库中的东东封装到了`std`名字空间中
- C++使用一套新的头文件, 这套头文件的文件名后不加`.h`扩展名

特别提醒

- ▶ 本课件部分例子采用**C**描述：

```
#include <stdio.h>
```

```
.....
```

```
scanf("%d", &n);
```

```
printf("Input n: ");
```

```
.....
```

- ▶ 大家可以自己修改为**C++**描述：

```
#include <iostream>
```

```
using namespace std;
```

```
.....
```

```
cin>>...;
```

```
cout<<...;
```

```
.....
```

Q & A

