



# 计算机程序 设计

# 第 1 讲 引言

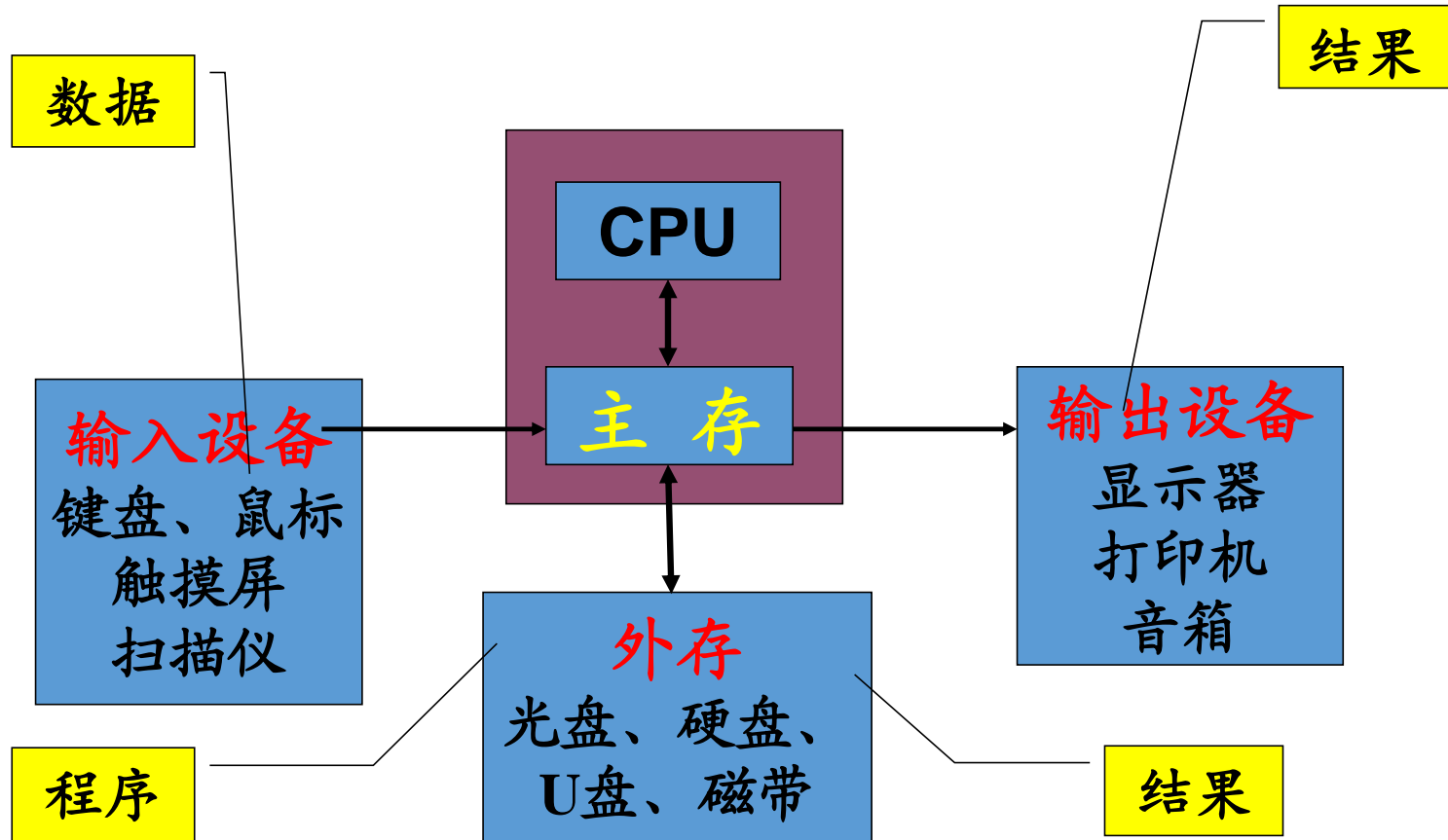
# 本章提要

- 计算机系统的工作原理
- 计算机上问题求解过程
- 计算机程序需要精巧设计
- 初识C语言
- 课程安排

# 1.1 计算机系统的工作原理

## 计算机是如何工作的？





# 内存

- 内存的组织——存储单元
- 内存的功能——数据存入和取出
- 存储的数据类型——二进制数据

# 二进制

- 1bit 0或1
- 1 Byte=8bits 00000000~11111111
- 1 KB(Kilobyte)=1024 bytes
- 1 MB(Megabyte)=1024 KB
- 1 GB(Gigabyte)=1024 MB
- 1 TB(Terabyte) = 1024 GB
- 1 PB(Petabyte) = 1024 TB
- 1 EB(Exabyte) = 1024 PB
- 1 ZB(Zettabyte) = 1024 EB

## 二进制数据解释

- 假设1Byte 内容为: 01000001。对该内容, 计算机可有多种不同的解释:
  1. **一个整数** — 01000001表示十进制整数65 , 直接进行二进制到十进制的转换,
  2. **一个字母** — 01000001编码表示 字符‘A’ , 计算机使用ASCII编码表示字符。
  3. **一条指令** — 01000001 也许告诉计算机两个数相加
- 如何解释一个比特流取决于计算机在做什么?



# 内存的组织

- 存储单元（1 byte）
- 存储单元地址
- 线性排列
- 存储的内容
  - 程序指令
  - 数据
  - 计算结果
  - 内存地址

地址	内容
0x1001	01000001
0x1002	
0x1003	
0x1004	
0x1005	
0x1006	
0x1007	
0x1008	

## 存储和取出

- 开机后，内存单元中每一位的状态并不是0，而是不可控制、随机的。
- **信息的存储：**重新设置存储单元的每一位为0或1。
- **信息的取出：**复制存储单元的内容到另一个存储位置，例如CPU的寄存器。

示例：加法指令

ADD 【0x1003】，【0x1007】，【0x1003】

• 一个可能程序执行过程如下：

- 程序开始执行后，首先存储数值10到存储单元【0x1003】中，然后存储数值20到存储单元【0x1007】中
- 整个指令的执行过程是读取存储单元【0x1003】的数值10到CPU的寄存器，再读取存储单元【0x1007】的数值20到CPU的寄存器。CPU计算结果为30。
- 将30存储到存储单元【0x1003】，即重置存储单元【0x1003】为数值30的二进制代码序列，而存储单元【0x1007】中的数值不变，仍然为20。

# 执行过程中内存图示

0x1003	11101101
0x1007	11101010

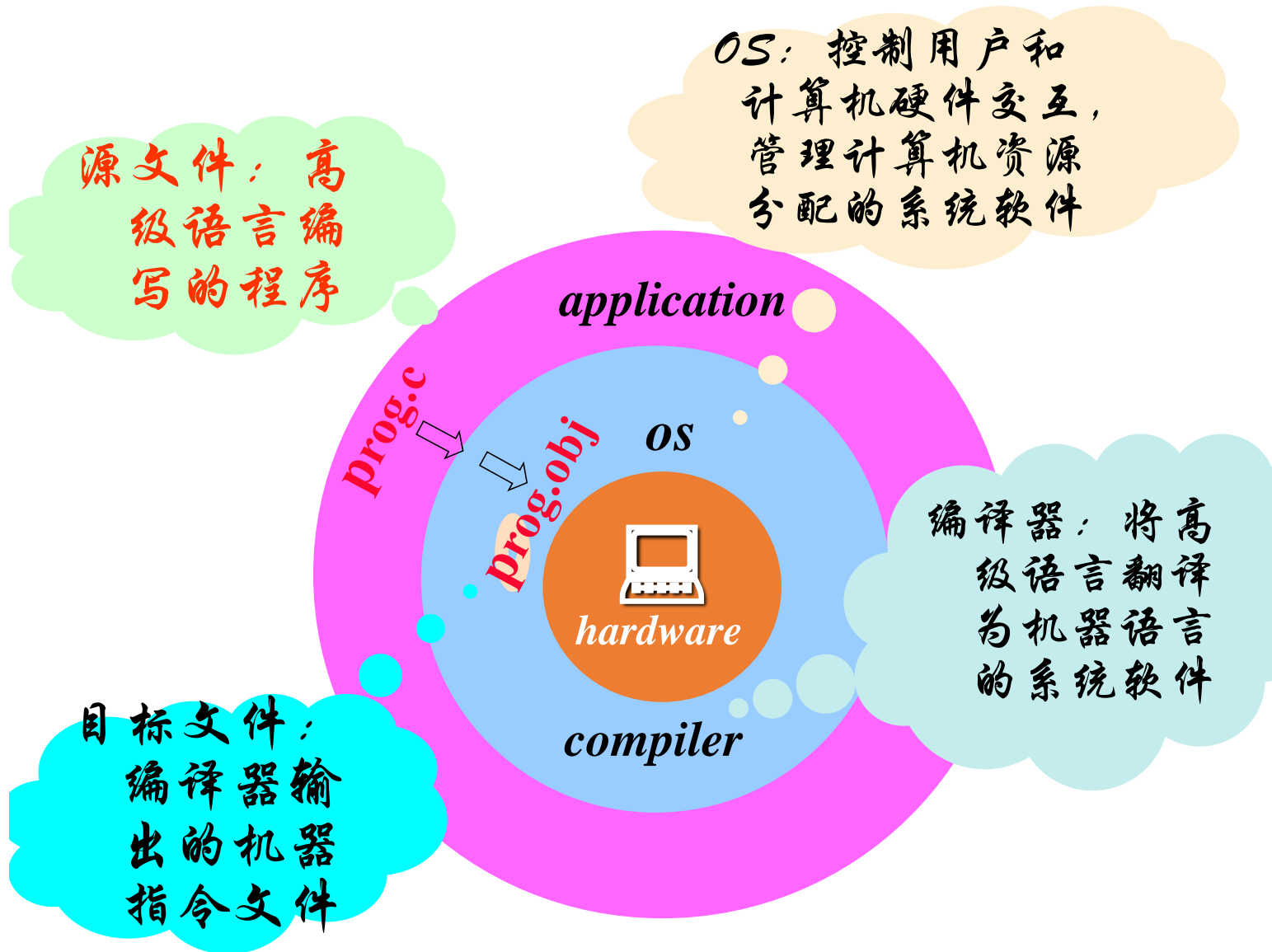
## 1. 初始状态

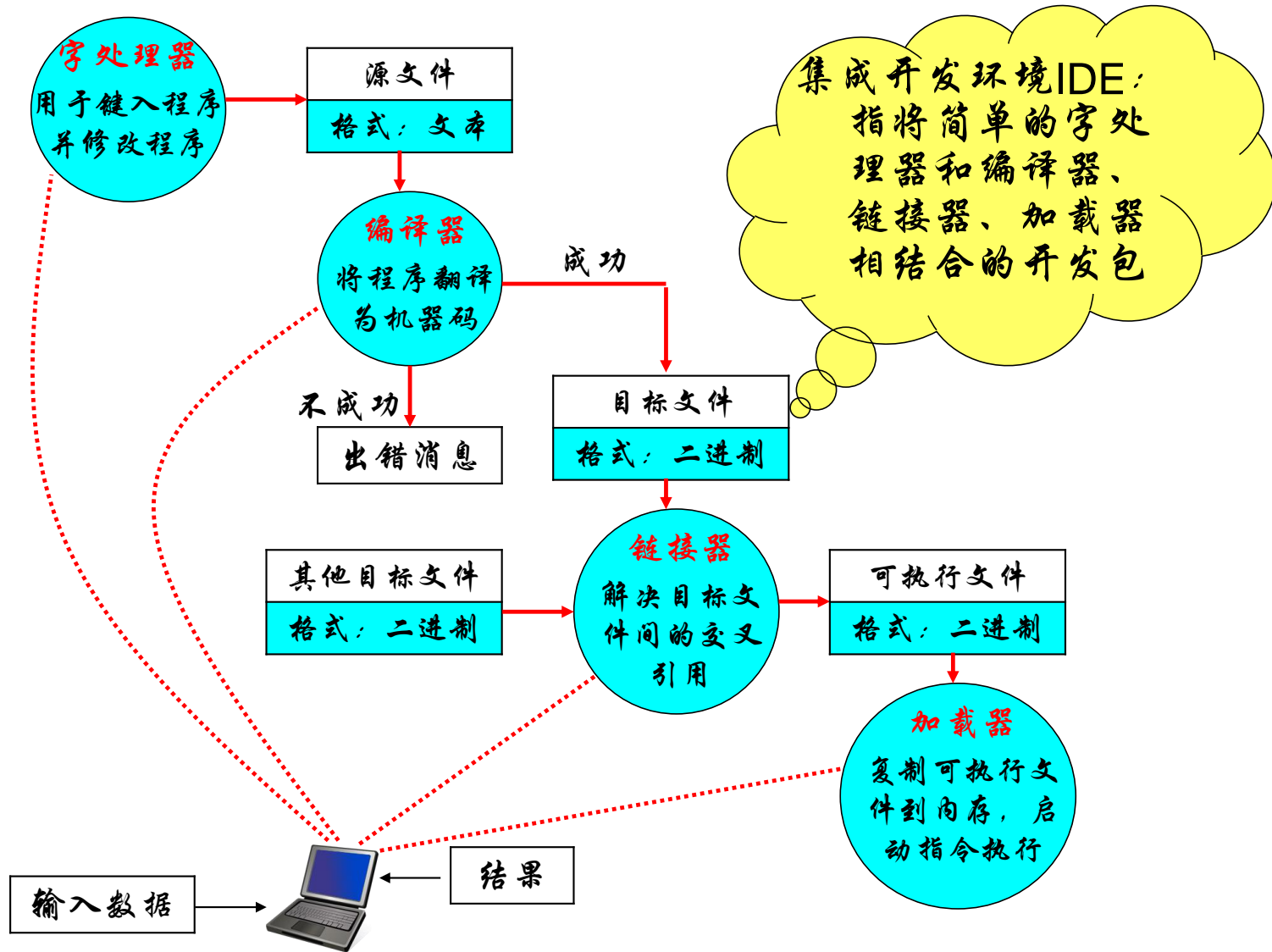
0x1003	00001010
0x1007	00010100

## 2. 赋值后的状态

0x1003	00011110
0x1007	00010100

## 3. CPU计算后的状态





## 1.2 计算机问题求解过程

### 步骤:

1. 分析问题
2. 设计解决问题的算法
3. 编程实现算法
4. 程序编译、调试
5. 运行

# 分析问题

- 分析问题的方法

- 输入
- 输出
- 解决问题的方法

- 数学中解答应用题

- 已知条件
- 求解对象
- 解题思路



# 设计算法

对同一个问题，可有不同的解题方法和步骤

广义地说，为解决一个问题而采取的方法和步骤，就称为“算法”。

例：求  $\sum_{n=1}^{100} n$

方法1：1+2，+3，+4，一直加到100 加99次

方法2：100+(1+99)+(2+98)+...+(49+51)+50  
= 100 + 49 × 100 + 50 加51次

- 为了有效地进行解题，不仅需要保证算法正确，还要考虑算法的质量，选择合适的算法。希望方法简单，运算步骤少。

- 计算机算法可分为两大类别：

- 数值运算算法：主要是管理与数据处理，如求数值解，求方程的根、求函数的定积分等
- 非数值运算算法：包括面十分广泛，如过程控制、定理证明等，其他最常见的是用于事务管理领域，如图书检索、人事管理、行车调度管理等

# 算法的特性

## ■ 一个算法应该具有以下特点：

- 有穷性：包含有限的操作步骤
- 确定性：算法中的每一个步骤都应当是确定的
- 有效性：算法中的每一个步骤都应当能有效地执行，并得到确定的结果，不存在无效的操作
- 有零个或多个输入：输入是指在执行算法时需要从外界取得必要的信息
- 有一个或多个输出：算法的目的是为了求解，“解” 就是输出

# 描述算法的工具

- 自然语言
- 流程图
- N-S流程图
- 伪代码
- ...

# 自然语言

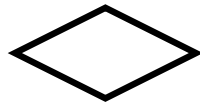
- 自然语言就是人们日常使用的语言，可以是汉语或英语或其它语言。用自然语言表示通俗易懂，但文字冗长，容易出现“歧义性”。自然语言表示的含义往往不大严格，要根据上下文才能判断其正确含义，描述包含分支和循环的算法时也不很方便。因此，除了那些很简单的问题外，一般不用自然语言描述算法

# 用流程图表示算法

- 美国国家标准化协会ANSI(American National Standard Institute)规定了一些常用的流程图符号：



起止框



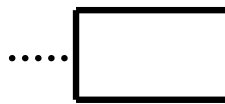
判断框



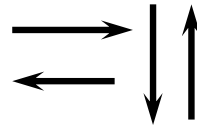
处理框



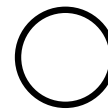
输入/输出框



注释框



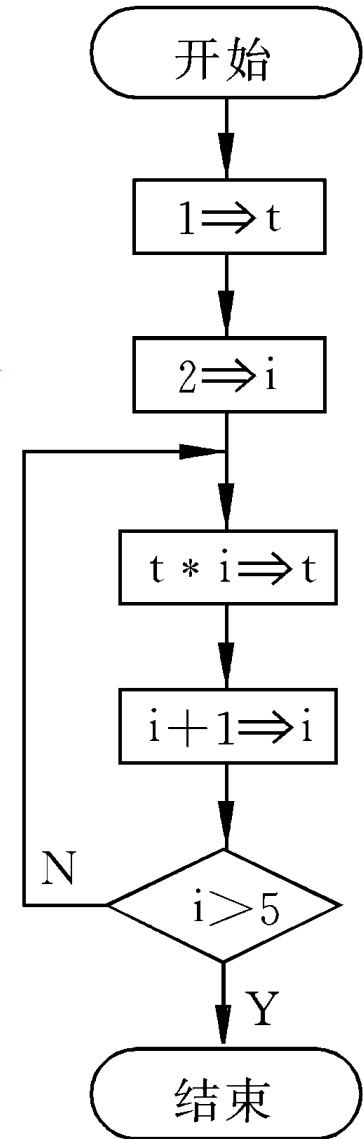
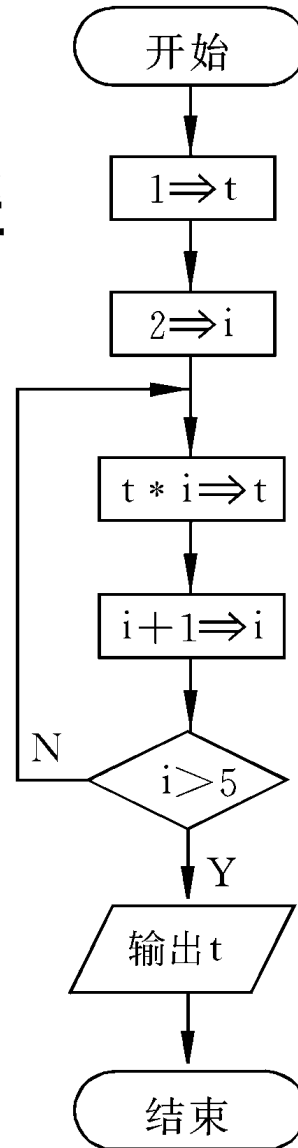
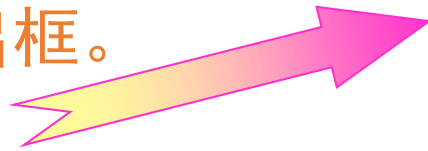
流向线



连接点

## 例:将求5!的算法用流程

如果需要将最后结果打印出来，可在菱形框的下面加一个输出框。



## 小结：

- 流程图是表示算法的较好的工具。  
一个流程图包括以下几部分：
- (1) 表示相应操作的框；
- (2) 带箭头的流程线；
- (3) 框内外必要的文字说明。



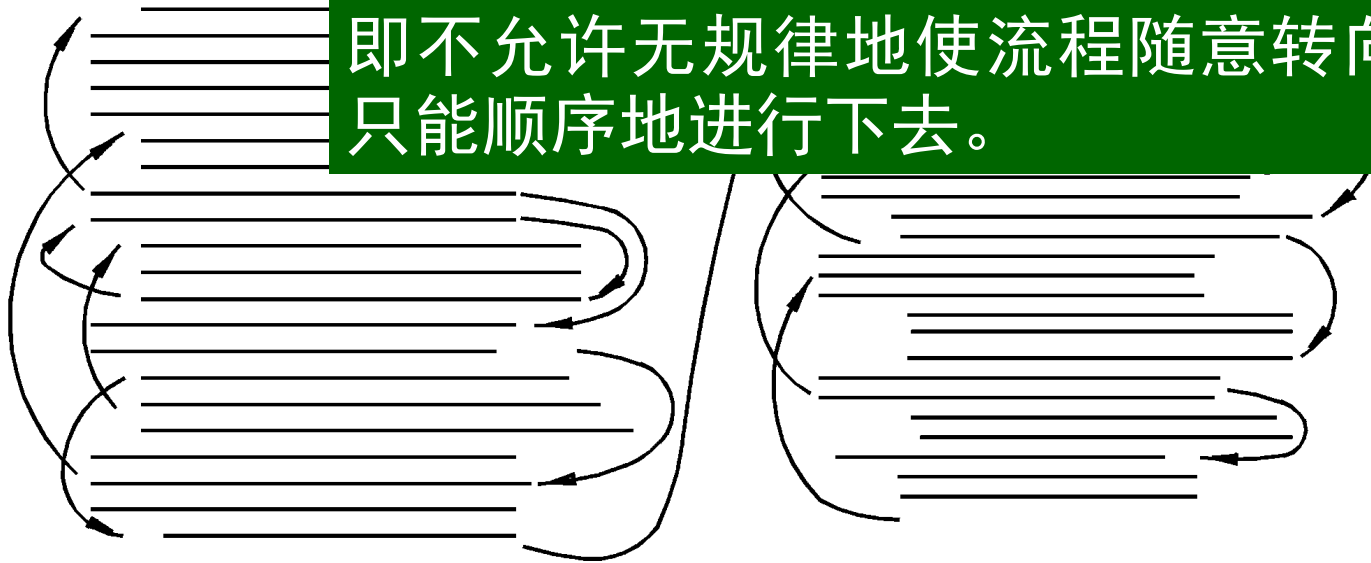
# 改进的流程图

## ■ 传统流程图的弊端

传统流程图用流程线指出各框的执行顺序，对流程线的使用没有严格限制。因此，使用者可以毫不受限制地使流程随意地转向，使流程图变得毫无规律，阅读者要花很大精力去追踪流程，使人难以理解算法的逻辑：



## 传统流程图



**缺点：**难以阅读、修改，使算法的可靠性和可维护性难以保证。

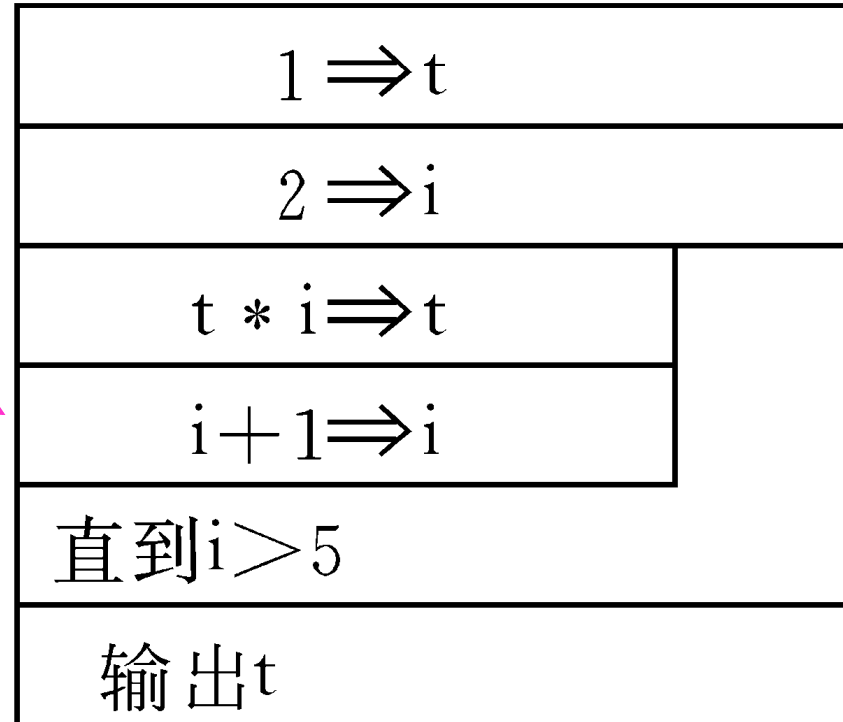
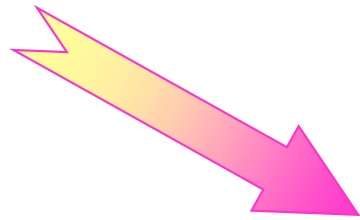
**解决办法：**必须限制箭头的滥用，即不允许无规律地使流程随意转向，只能顺序地进行下去。

这种如同乱麻一样的算法称为BS型算法，意为一碗面条(A Bowl of Spaghetti)，乱无头绪。

## 用N-S流程图表示算法

1973年美国学者I. Nassi和B. Shneiderman提出了一种新的流程图形式。在这种流程图中，完全去掉了带箭头的流程线。全部算法写在一个矩形框内，在该框内还可以包含其它的从属于它的框，或者说，由一些基本的框组成一个大的框。这种流程图又称N-S结构化流程图。

**例：求5!算法  
用N-S图表示**



## N-S图表示算法的优点

- 比文字描述直观、形象、易于理解；比传统流程图紧凑易画。尤其是它废除了流程线，整个算法结构是由各个基本结构按顺序组成的，N-S流程图中的上下顺序就是执行时的顺序。用N-S图表示的算法都是结构化的算法，因为它不可能出现流程无规律的跳转，而只能自上而下地顺序执行。

# 用伪代码表示算法

- 概念：伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。
- 特点：它如同一篇文章一样，自上而下地写下来。每一行(或几行)表示一个基本操作。它不用图形符号，因此书写方便、格式紧凑，也比较易懂，也便于向计算机语言算法（即程序）过渡。
- 用处：适用于设计过程中需要反复修改时的流程描述。

例：用伪代码表示求  
5!算法：

也可以写成以下形式：

BEGIN { 算法开始 }

1 → t

2 → i

while i ≤ 5

{ t × i → t

i + 1 → i }

print t

END { 算法结束 }

开始

置t的初值为1

置i的初值为2

当 i ≤ 5，执行下面操作：

使 t = t × i

使 i = i + 1

{ 循环体到此结束 }

输出t的值

结束

# 编写代码

## • 算法的程序实现

- 变量定义
- 数据输入
- 程序语句
- 结果输出

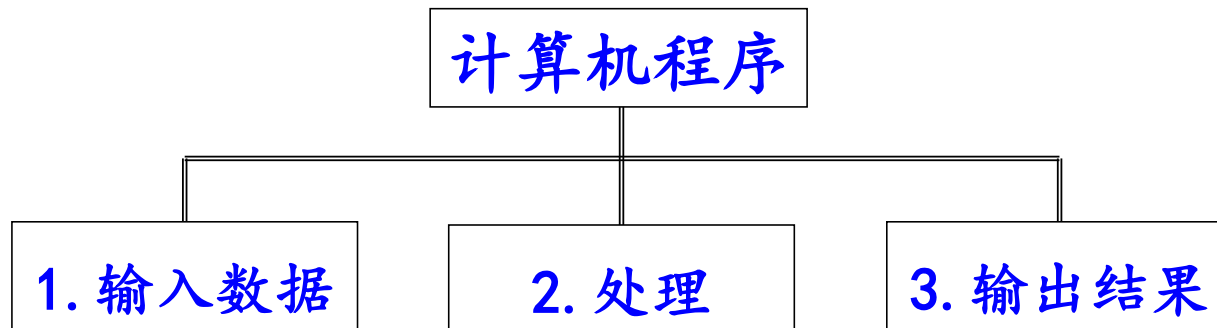
## • 解决问题的算法

- 数学符号
- 获取数据
- 求解步骤
- 问题结果



## 最基本的程序结构

- 只包含一个模块
- IPO ( **I**ntput、**P**rocessing、**O**utput ) 结构：  
由IBM公司发起，考虑一个模块的基本结构
- IPO结构中O是必须的，IP则根据功能需求存在



# 运行求解

- 编译程序
- 链接资源
- 运行程序
- 分析结果

## 案例：华氏温度到摄氏温度的转换

美国及许多国家使用华氏温度为单位，而中国使用摄氏温度为单位，现在希望编写程序以实现华氏温度到摄氏温度的转换。

# 分析问题

- 问题的输入

华氏温度  $f$

- 问题的输出

摄氏温度  $c$

- 解决问题的方法

$$\text{摄氏温度} = \frac{5}{9} * (\text{华氏温度} - 32)$$

# 设计算法

1. 获得华氏温度的数值
2. 将华氏温度转换成摄氏温度
3. 显示摄氏温度的数值

# 编写代码

## •数学符号

- 表示华氏温度的变量
- 表示摄氏温度的变量

## •数据类型选择

- 浮点数据

## •功能

- 获得以华氏度表示的温度
- 将华氏度转换成摄氏度
- 输出以摄氏度表示的温度

```
/* 华氏温度到摄氏温度的转换 */  
#include <stdio.h>  
int main ( )  
{  
    float f, c;    // 定义f和c为单精度浮点型变量  
    f=32.;        // 指定f的值  
    c=(5.0/9)*(f-32);    // 利用公式计算c的值  
    printf("f=%f\nc=%f\n", f, c);    // 输出c的值  
    return 0;  
}
```

## 运行解题

```
f=32.000000  
c=0.000000
```

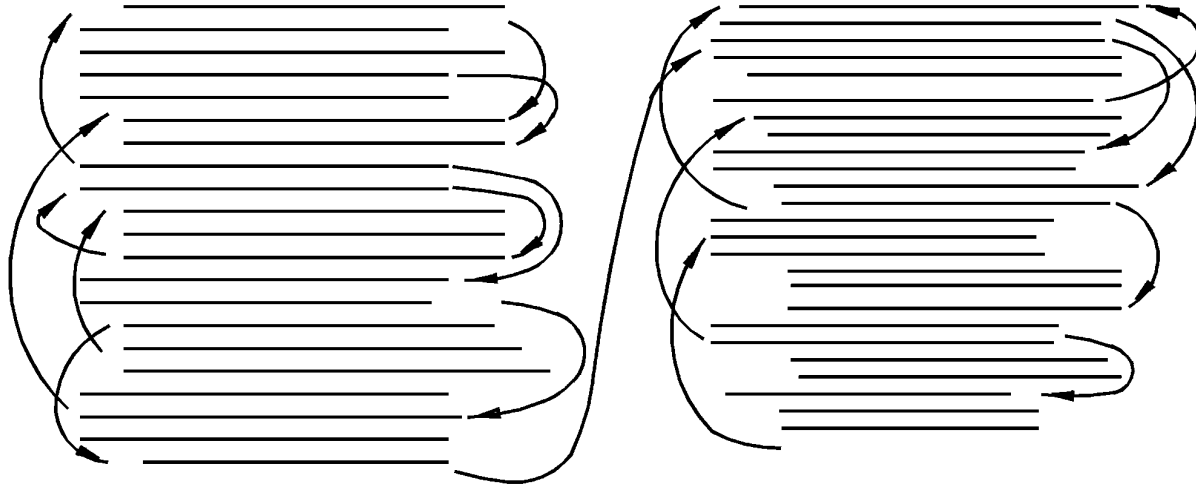
```
f=68.000000  
c=20.000000
```



## 1.3 计算机程序需要精巧设计

### 计算机程序

- **程序**：一组计算机能识别和执行的**指令**。程序在计算机中是以0、1组成的指令代码来表示的
- 只要让计算机执行这个程序，计算机就会**自动地、有条不紊地**进行工作。计算机的一切操作都是由**程序**控制的，离开程序，计算机将一事无成
- 计算机所能实现的指令的集合称为计算机的**指令系统**。指令个数是有限的，且每条指令只能完成计算机的一个最基本的功能，但一系列指令的组合却能完成很复杂的功能



- **优秀的程序**：逻辑清晰，代码冗余少，可读性好，健壮性、可移植好，内存占用量小，执行时间短...
- 优秀的程序哪里来？ **设计**

# 如何设计计算机程序？



程序=数据结构+算法      N.Wirth(沃斯)

- 设计程序：选择合适的数据结构与算法
- 程序设计方法学
  - 用以指导程序设计各阶段工作的原理和原则，以及依此提出的设计技术
  - 目标：设计出可靠、易读而且代价合理的程序
  - 基本内容：结构程序设计；程序理论在程序设计技术中的应用，规格说明和变换技术

# 程序设计语言

- 也叫**计算机语言**：人和计算机交流信息的、计算机和人都能识别的语言

- **计算机语言发展阶段**：



低级语言

- **机器语言**（由0和1组成的指令）
- **符号语言**（用英文字母和数字表示指令）
- **高级语言**（接近于人的自然语言和数学语言）
  - 面向**过程**的语言  
（非结构化的语言、结构化语言）
  - 面向**对象**的语言

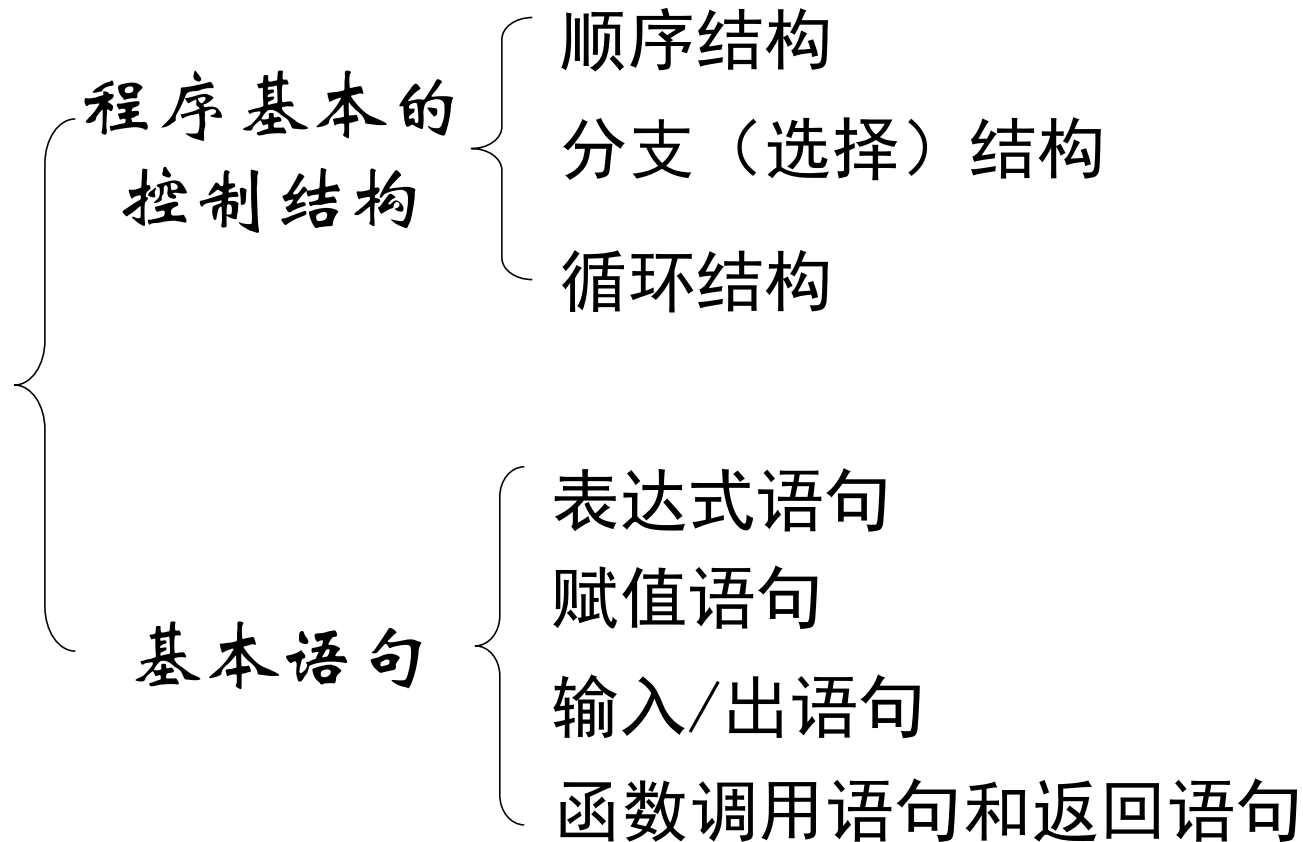
- 程序设计语言必须具备数据表达和流程控制两大功能。

## 数据表达:

- 客观世界通过数据来描述，一类数据具有的共同特点就是数据类型。
- 数据类型定义的内容
  - 该类型的数据是什么
  - 能在这些数据上做什么
- 数据类型
  - 基本数据类型——整型、实型、字符型
  - 构造类型——数组、结构、文件、指针

## 流程控制

- **结构化程序设计**——复杂任务划分为子任务。每个子任务独立编程，最后通过积木式的扩展完成复杂任务
- 程序的控制过程通过程序中的一系列语句来实现。



# 初识C语言

## 为什么是C语言？最好的入门语言

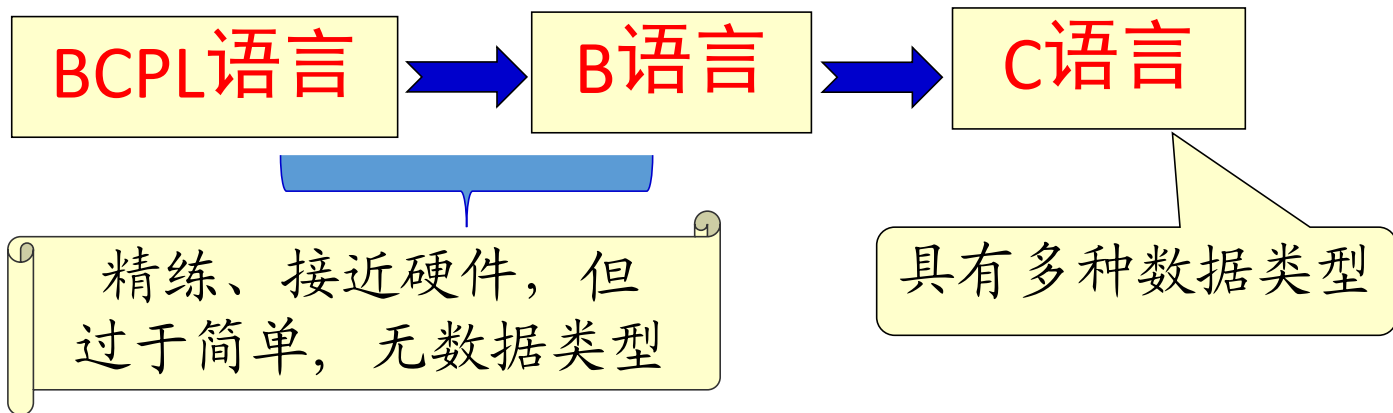
- C语言是国际上广泛流行的计算机高级语言
- 对于想从事嵌入式底层开发以及对开源操作系统特别是Linux感兴趣的同学，学好C语言是必须具备的功课，底层操作系统实现语言基于C语言或者汇编，绝大部分是C
- 对于有志于了解整个计算机系统架构或者做架构师的同学，C语言是必须要深刻掌握的语言，它比C++、Java、Python语言等更接近机器



- C语言作为存在历史这么长久的语言，基于C语言的架构体系留存在各个大公司，如果掌握了C语言工作机遇多
- 如果具备了C语言基础再去学习更加高级的语言，将相对容易
- 开源代码很多都是基于C语言实现。加入开源社区可以学习标准的代码规范，了解最前沿的技术发展方向，也可以利用开源项目，提升自身的项目经验

## C语言的发展:

- C语言是在B语言的基础上发展起来的，其前身可以追溯到 **ALGOL 60**(1960)、**CPL**(1963)、**BCPL**(1967)
- 1970年贝尔实验室的Ken Thompson进一步简化了BCPL，设计了**B语言**，并用于编写了第一个UNIX操作系统



- 1972年，贝尔实验室的Dennis Ritchie 在B语言的基础上设计了C语言，对B取长补短
- 1973年，Ken Thompson和Dennis Ritchie用C改写了原来用汇编编写的UNIX，（即UNIX第5版），但仅在贝尔实验室使用

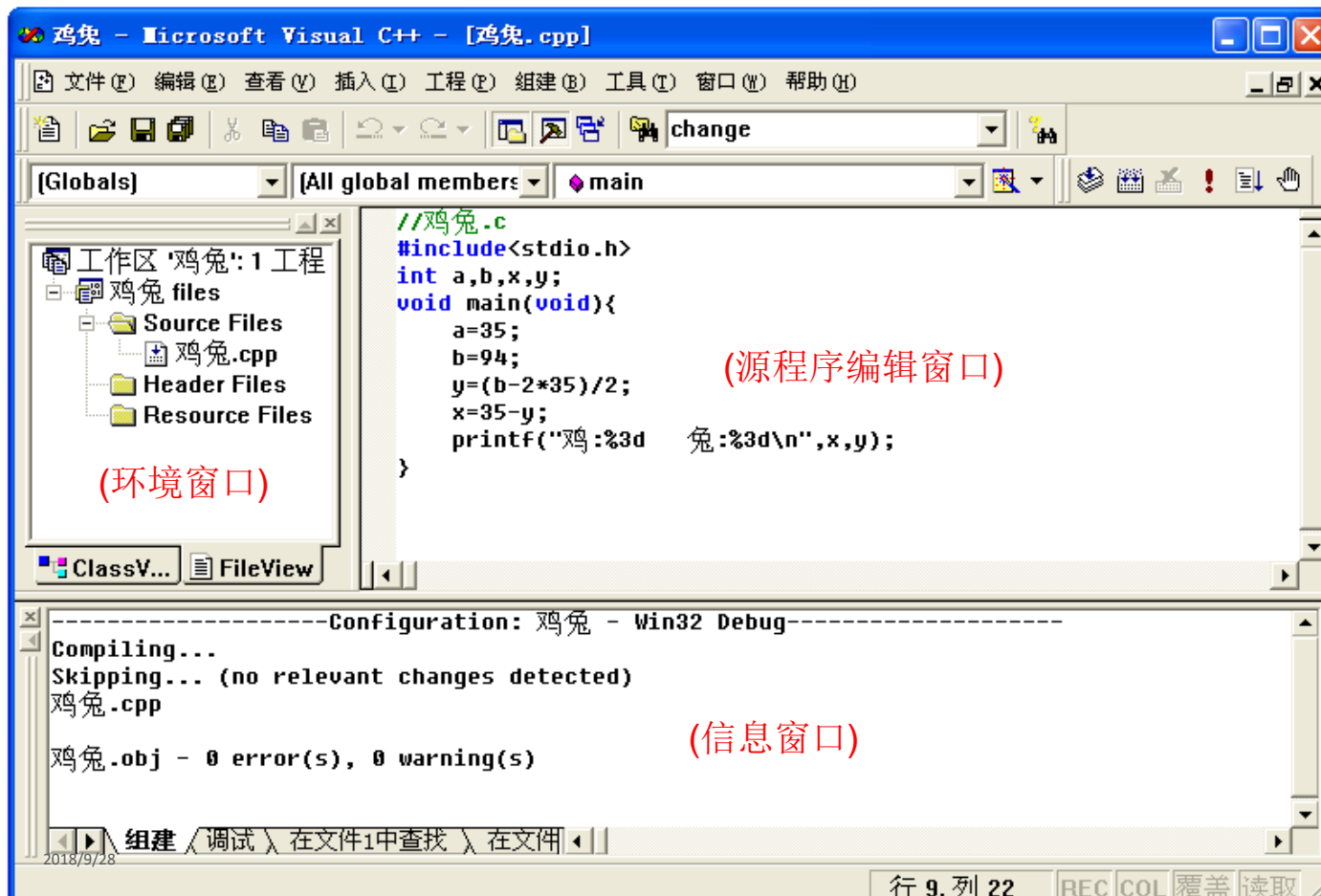


- 1975年UNIX第6版发布，C优点突出引起关注。
- 1977年出现了《可移植C语言编译程序》，推动了UNIX在各种机器上实现，C语言也得到推广，其发展相辅相成。
- 1978年影响深远的名著《**The C Programming Language**》由 Brian W. Kernighan和Dennis M. Ritchie 合著，被称为标准C。
- 之后，C语言先后移植到大、中、小、微型计算机上，已独立于UNIX和PDP，风靡世界，成为最广泛的几种计算机语言之一。

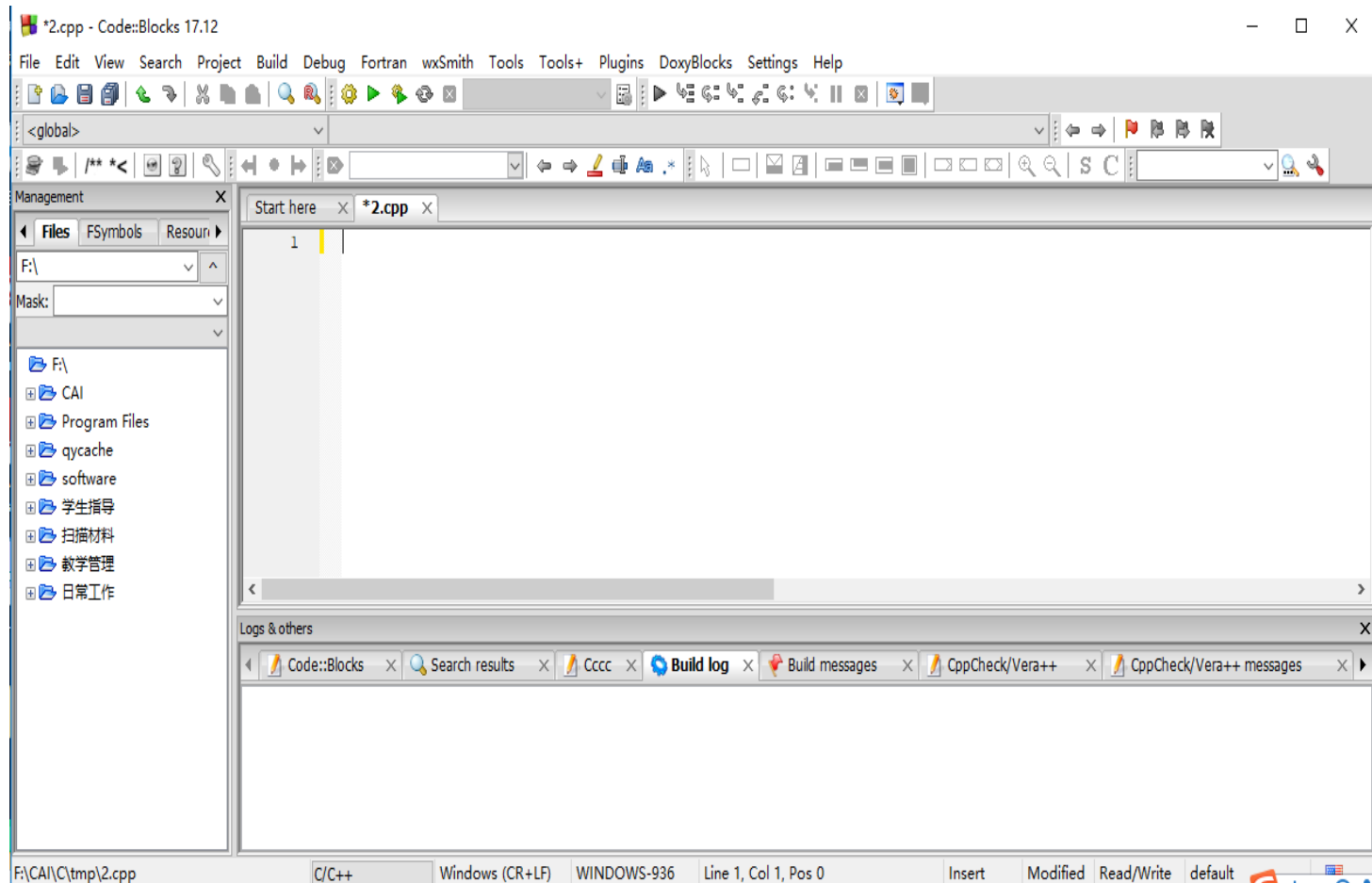
- 1983年，美国国家标准化协会ANSI根据C语言各种版本对C的发展和扩充，制定了新的标准ANSI C，比标准C有了很大的发展
- 1988年K & R按照 ANSI C修改了他们的《The C Programming Language》
- 1989年，ANSI公布了一个完整的C语言标准—ANSI X3.159-1989(常称ANSI C，或C89)

- 1990年，国际标准化组织接受C89作为国际标准ISO/IEC 9899:1990，它和ANSI的C89基本上是相同的
- 1995年，ISO对C90作了一些修订（C95）
- 1999年，ISO又对C语言标准进行了更大的修订，在基本保留原来的C语言特征的基础上，增加了一些功能，尤其是C++中的一些功能，命名为ISO/IEC 9899:1999，2001、2004年先后进行了两次技术修正。ISO/IEC 9899:1999(及其技术修正)被称为 **C99**

# Visual C++6.0 集成开发环境

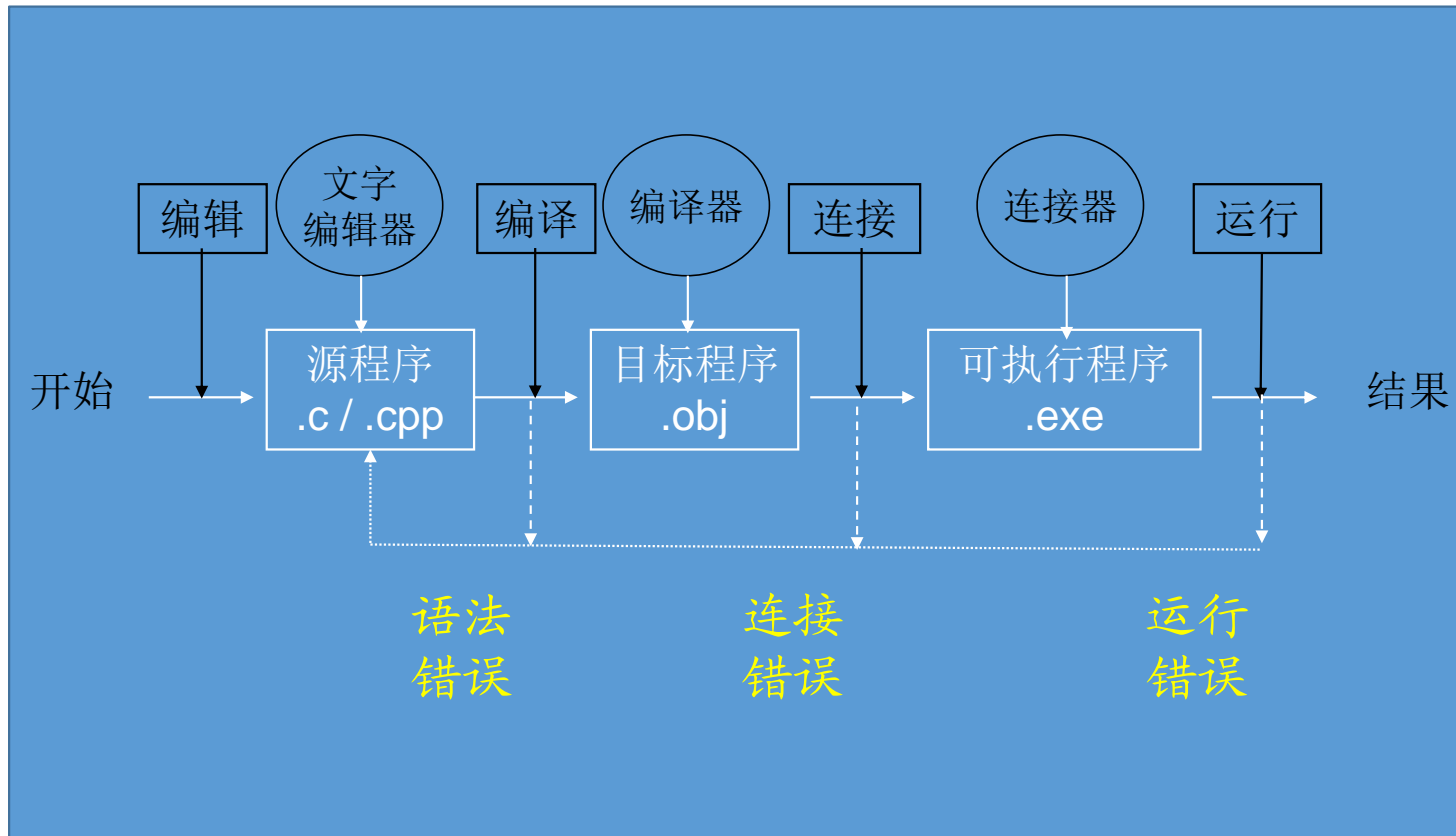


# Code::Blocks 17.12集成开发环境





## C语言程序的编辑、调试、运行步骤



后缀名	解释
cpp或c	源程序文件
dsw	项目工作区
dsp	项目文件
ncb	无编译浏览文件
plg	编译信息文件
opt	关于开发环境的参数文件，如工具条位置等信息

## 简单的C语言程序举例

要求在屏幕上输出以下一行信息

This is a C program.

➤ **解题思路：**

在主函数中用**printf**函数原样输出以上文字。

```
#include <stdio.h>                                /*文件包含*/
int main ( )                                       /*主函数 */
{                                                  /*函数体开始*/
    printf ("This is a C program.\n");          /*输出语句*/
    return 0;
}                                                  /*函数体结束*/
```

说明: main-主函数名, void-函数类型

- 每个C程序必须有一个主函数main
- { }是函数开始和结束的标志, 不可省
- 每个C语句以分号结束
- 使用标准库函数时应在程序开头一行写:

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

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

```
int main( )
```

C程序必须有一个 main 函数

{ 函数的名字，表示主函数

```
printf ("This is a C program.\n");
```

```
return 0;
```

```
}
```

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

```
int main( )
```

```
{
```

主函数类型

```
    printf ("This is a C program.\n");
```

```
    return 0;
```

```
}
```

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

```
int main( )
```

```
{
```

```
    printf ("This is a C program.\n");
```

```
    return 0;
```

```
}
```



函数体

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

```
int main( )
```

```
{
```

输出函数

```
printf ("This is a C program.\n");
```

---

```
return 0;
```

输出语句

```
}
```



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

```
int main( )
```

```
{
```

```
    printf ("This is a C program.\n");
```

---

```
    return 0;
```

```
}
```

输出语句

```
This is a C program.  
Press any key to continue_
```

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

```
int main( )
```

```
{
```

```
    printf ("This is a C program.\n");
```

```
    return 0;
```

```
}
```

A screenshot of a terminal window with a black background and white text. It displays two lines of output: "This is a C program." followed by a newline, and "Press any key to continue\_" followed by a cursor. The text is in a monospaced font.

This is a C program.  
Press any key to continue\_

A yellow callout box with a black border and a pointer directed at the '\n' character in the printf statement. It contains the Chinese text "换行符" (newline character) in red.

换行符

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

```
int main( )
```

```
{
```

```
    printf ("This is a C program.\n");
```

```
    return 0;
```

```
}
```

当main函数执行结束前  
将整数0作为函数值

#include<stdio.h>

int main( )

用到函数库中的输入输出函数时

{

printf ("This is a C program.\n");

return 0;

}

表示语句结束

## 中国古代数学“鸡兔同笼”问题

“今有鸡兔同笼，上有三十五头，下有九十四足，问鸡兔各几何”

设有a只鸡，b只兔

$$a + b = 35 \dots\dots\dots(1)$$

$$2a + 4b = 94 \dots\dots\dots(2)$$

## 1. 分析问题

分析问题需要明确输入，也就是要处理的数据；明确输出，也就是结果；解决方案的任何附加需求或约束条件。

问题输入：总脚数94，总头数35

问题输出：鸡数a和兔数b

关系公式：  $a + b = 35$

$$2a + 4b = 94$$

推导公式：

$$b = (94 - 2 \times 35) \div 2$$

$$a = 35 - b$$

## 2. 算法流程图

鸡兔同笼问题

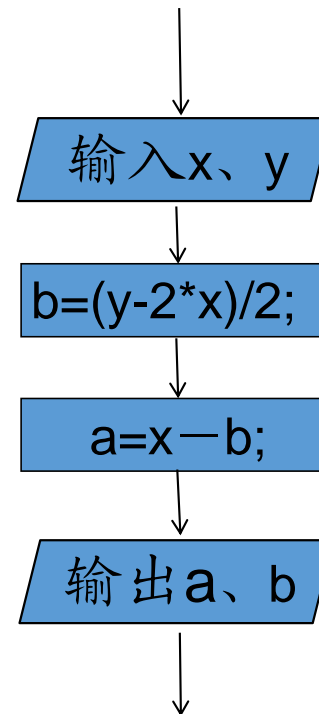
假设：

**x**代表总头数，

**y**代表总脚数，

**a**鸡数，

**b**兔数



### 3.编程实现算法

```
//程序名：鸡兔.c 主要功能：计算鸡兔数
#include<stdio.h> //引入头文件
int x, y, a, b;    //声明4个整型变量
void main() {      //主函数
    printf("Please input total head number (x) and
           total foot number (y):\n");
    scanf("%d%d", &x, &y);           //输入数据
    b=(y-2*x)/2;
    a=x-b;
    printf("鸡:%3d 兔:%3d\n", a, b); //输出结果
}
```



# 课程学习安排

## 学习目标

1. 掌握程序设计的思想、方法和技术
2. 培养问题求解能力
3. 提高C语言编程能力
4. 最终达到
  - 专业的程序设计能力👉
  - 良好的程序设计风格👉
  - 富有创造力的算法和技巧👉

# 学习方法

- **课前**——预习，带着问题来听课
- **课堂**——认真听讲，积极发言，重在理解
- **课后**——总结方法，多读程序，重视编程实践
- 参考书列表
  - **C语言程序设计（第3版），何钦铭、颜晖，高等教育出版社**
  - **C语言程序设计实验与习题指导（第3版），颜晖、柳俊，高等教育出版社**
  - **Engineering Problem Solving with C（Fourth Edition），Delores M. Etter**

# 课程考核方法

- 期终考试60%（笔试闭卷）
- 平时成绩40%（平时统计）
  - 实验报告（20%）
  - 上机+考勤+提问（20%）
- 三次缺勤平时成绩按0分计
- 总成绩网上自动生成

# Homework

复习教材第1章；

熟悉 Visual c++6.0 集成开发环境：《教材》  
附录A、《学习辅导》 p223-237

作业：《教材》 p14    3、5、6、7  
                                p35    4、5、6

实践：《教材》 p14    5、6、7；  
          《学习辅导》： p253 实验1

课程网站：172.18.113.22（谷歌浏览器）

用户名：学号

初始密码：学号