



广东岭南职业技术学院
GUANGDONG LINGNAN
POLYTECHNIC

现代制造学院 | 香港铸业学院

C 语言



教师：黄国爵
QQ：651308462





授课主题

授课主题的详细内容

教学目标

教学目标的详细内容

教学重点、难点与解决方案

重点：

难点：

学时分配



目录

01

前言

02

C语言概述

03

编写简单的 C 语言程序

04

算法





01

PART ONE

前言





学习内容

- 基础知识，语法规则
- 程序设计方法（算法分析与设计）
- 上机操作、程序调试
- 阅读程序的技巧
- 写程序的技巧





常用编程软件

Turbo-C 3.0

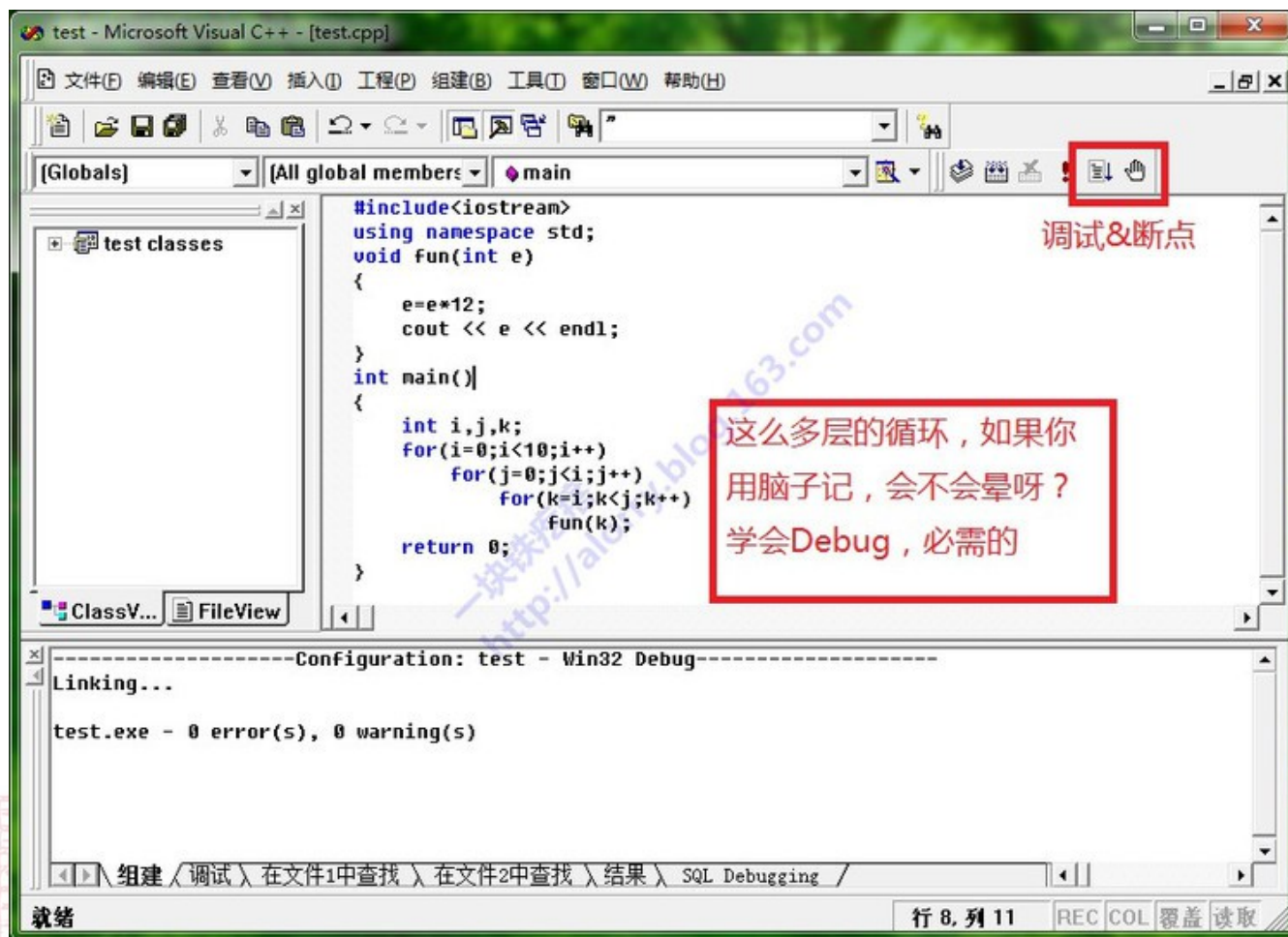




常用编程软件



Visual Stdio (VC++ 6.0)





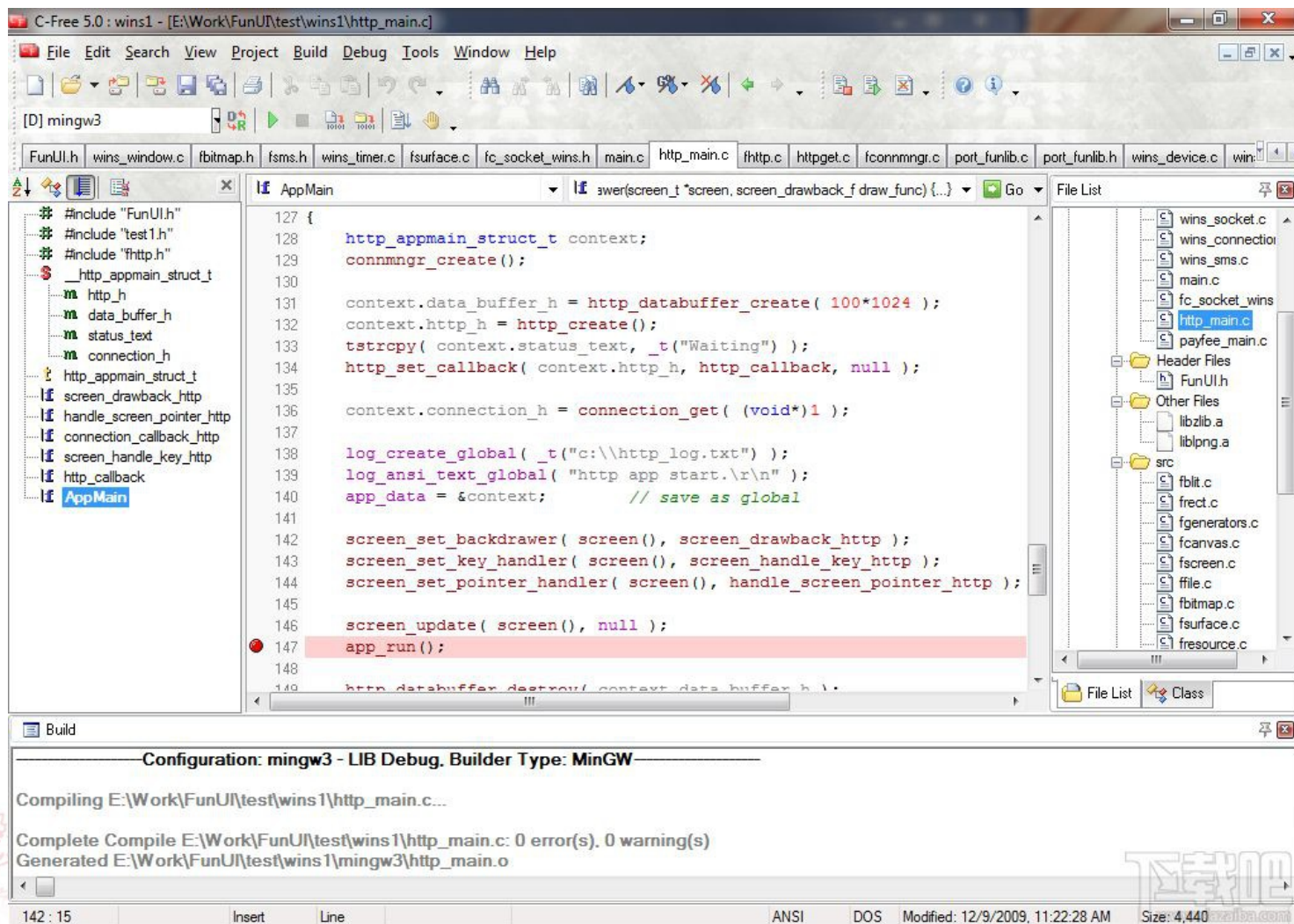
常用编程软件



C-Free

Start Here

C-free

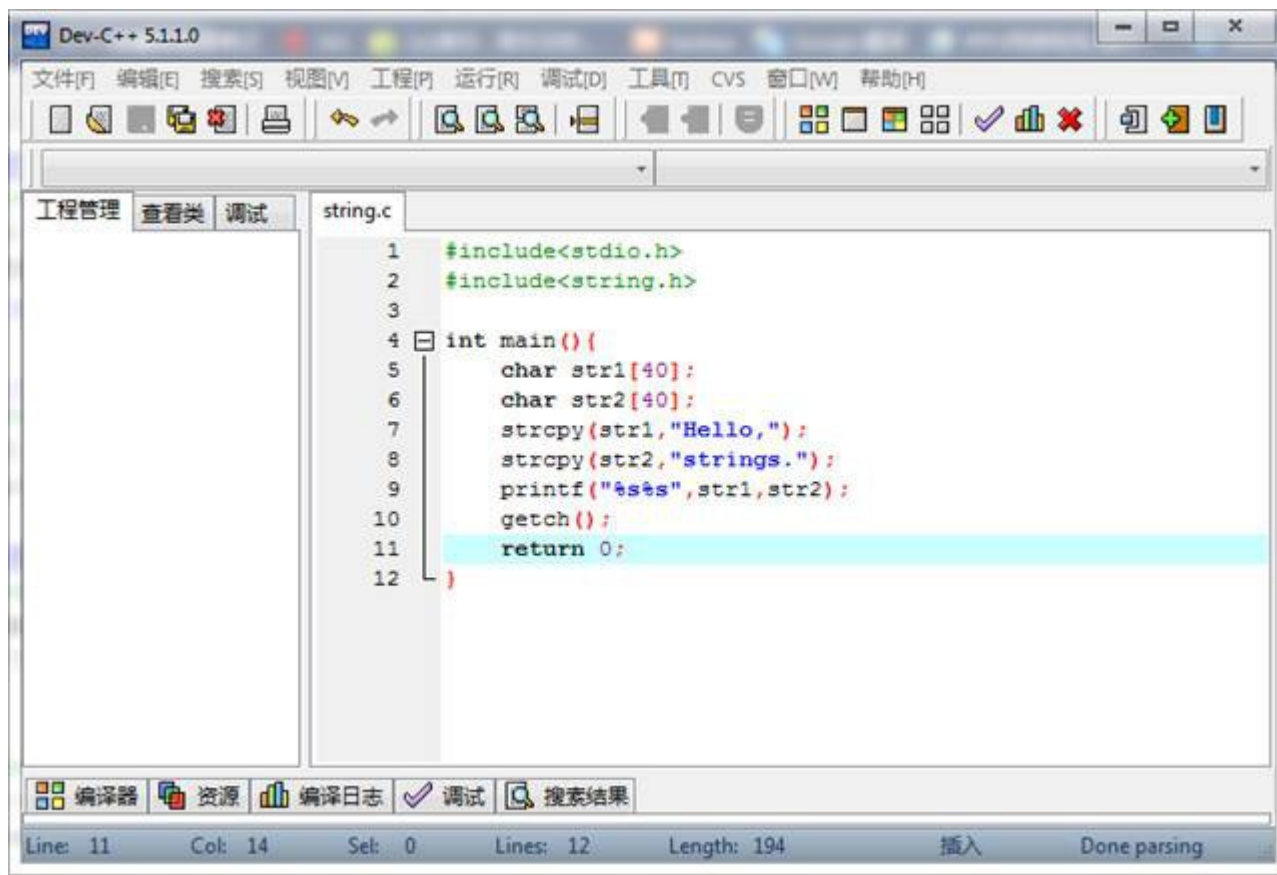




常用编程软件



Dev-C++





学时分配

- 36 学时授课
- 36 学时上机





考核方式：

平时成绩（40%）

- —— 作业、考勤
- —— 课堂纪律

期末考试（60%）

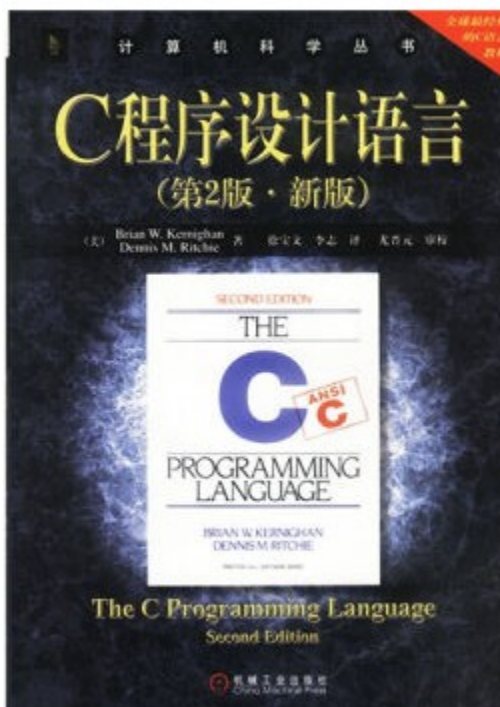
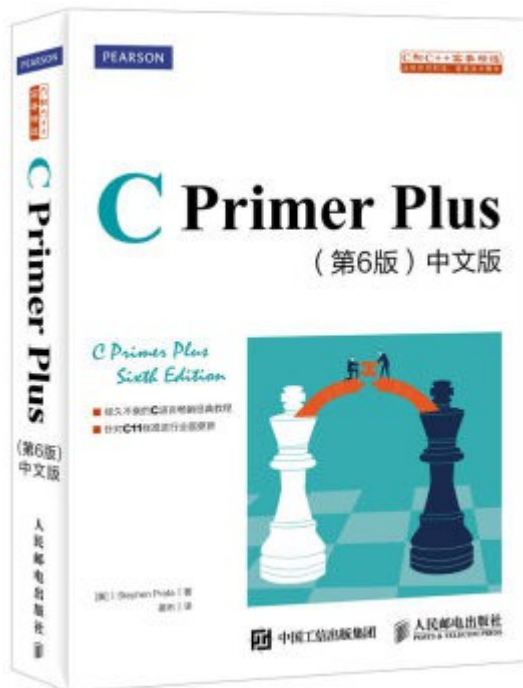
—— 开卷考试





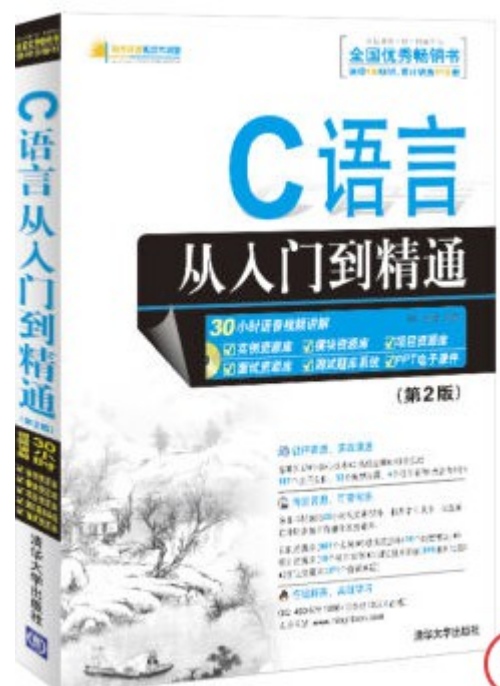
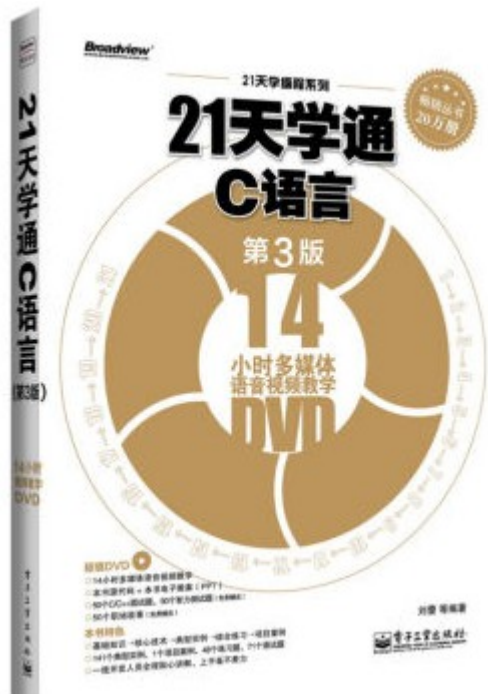
课内参考书目

- 《C Primer Plus 第6版 中文版》
- 《C 程序设计语言（第2版·新版）》
- 《C 程序设计（第三版）》谭浩强



课内参考书目

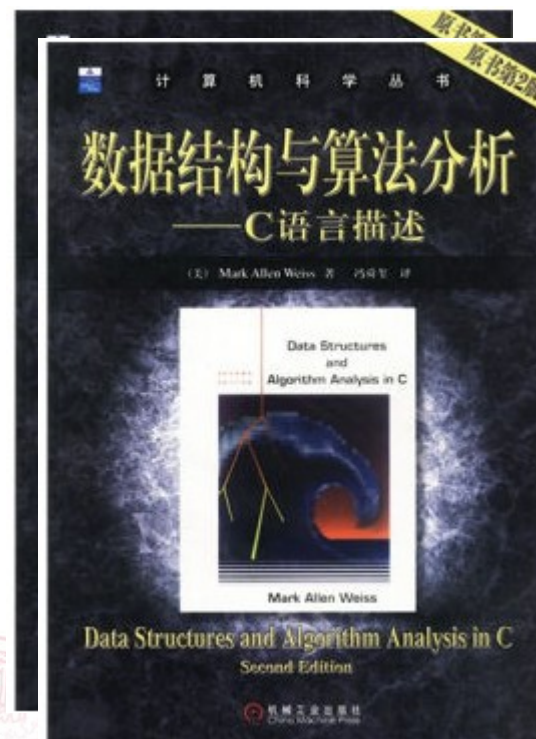
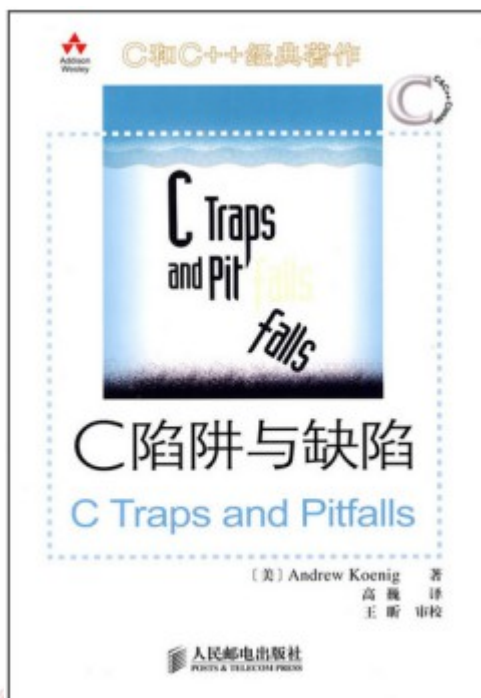
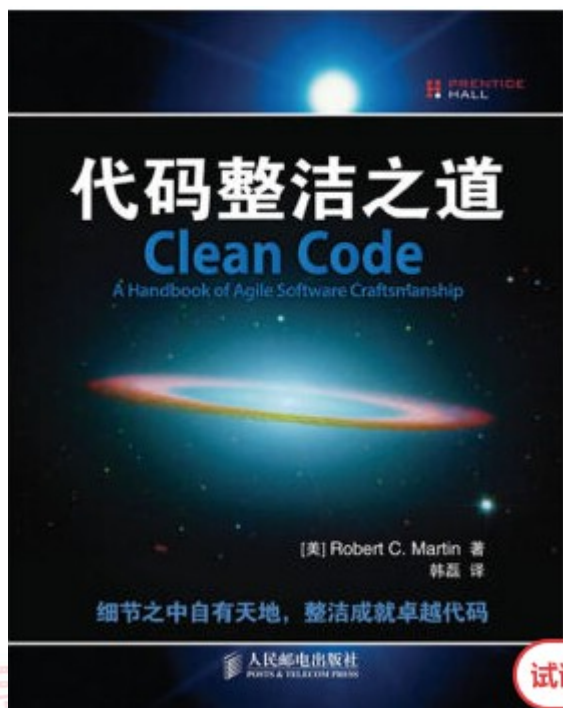
- 《21天学通C语言（第3版）》
- 《C语言从入门到精通（第2版 附光盘）》





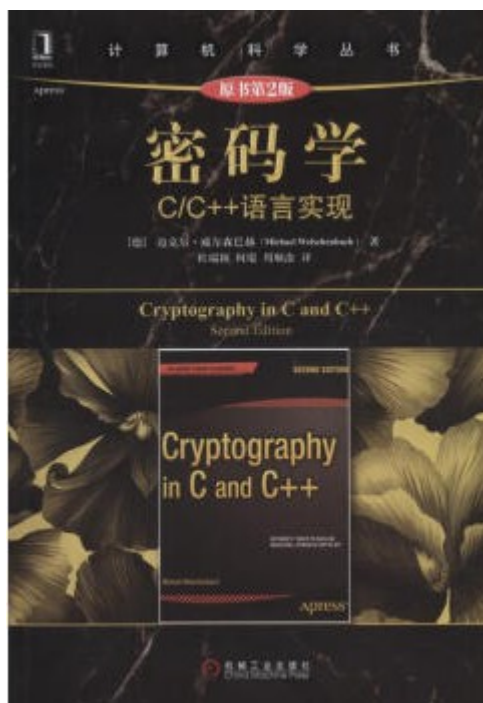
进阶参考书目

- 《代码整洁之道》
- 《C 陷阱与缺陷》
- 《数据结构与算法分析：C 语言描述》



进阶参考书目

- 《编译原理》
- 《密码学： C/C++ 语言实现》





02

PART TWO

C 语言概述





C 语言概述

- C 语言发展史
- C 语言程序组成简介
- C 语言开发到执行的过程
- C 语言特点
- C 语言开发环境介绍





一、C语言的发展





C语言的发展：

- C语言诞生于20世纪70年代，最初是用于PDP-11计算机上实现UNIX操作系统。
- 1977年出现了独立于机器的C语言编译文本。1978年贝尔实验室正式发表了C语言。
- 1983年ANSI为C语言制定了ANSI C标准，并于1988年完成。
- 现今C语言已成为世界上应用最广泛的计算机语言之一。





二、C 语言程序的结构





2.1、几个简单的例子

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

预处理命令

```
void main ( )
```

```
{
```

主函数

```
printf ( "Hello World ! \n" ) ;
```

```
}
```

函数体

程序运行的输出结果：

Hello World !





```
#include <stdio.h>
main ( )
{
    int a, b, sum;
    a=134;b=258;
    sum=a+b;
    printf ( “sum is  %d  \n” , sum);
}
结果:  sum is 392
```





```
main( )  
{  
    int a,b,c;  
    scanf("%d,%d", &a,&b);  
    /*  
    c=max(a,b);  
    /  
    printf("max=%d",c);  
}
```

/* 主函数 */

/* 定义变量 */

/* 输入变量 a 和 b 的值

/* 调用 max，值赋给 c*

/* 输出 C 的值 */





```
int max(int x, int y)    /* 定义 max 函数, x,
y 参数 */
{
    int z;                /* 声明部分, z 为整
    型 */
    if (x>y) z=x;
    else z=y ;
    return(z);            /* 将 z 通过 max 带回
    调用处 */
}
```

输入: 8 , 5
输出: max=8





2.2、C程序的基本结构

2.2.1 函数

函数是构成 C 语言的基本单位，主要有以下几种类型：

- 1、main 函数：是任何一个 C 语言程序开始执行的地方
一个 C 语言程序至少要包含一个函数 ---main 函数。

如果没有 main() 函数，程序提示出错。（现场例子）

main 是主函数的函数名，表示这是一个主函数。

每一个 C 源程序都必须有，且只能有一个主函数（main 函数）。



2、 用户自定义函数

由用户自定义，如上例中的 `max()` 函数。

函数（除主函数外）可以被其他函数调用

函数的好处？保持程序的结构性，可共享。





3、系统函数

函数调用语句，printf 函数的功能是把要输出的内容送到显示器去显示。

printf 函数是一个由系统定义的标准函数，可在程序中直接调用。

系统函数由系统内含，不过在使用时必须带上包含该函数说明的头文件。如：

scanf、printf 函数在头文件 `stdio.h` 中声明

sqrt、cos 函数在头文件 `math.h` 中声明





4、函数的基本形式

1) 函数的首部

返回值类型 函数名 (参数列表)

如: int max (int x, int y)

函数名后必须带 (), 即使参数列表为空

2) 函数体

函数体指首部后紧接的大括号部分 { }

函数体一般包括两个部分:

声明部分: 定义这部分所要用到的变量。

执行部分: 由若干语句组成。





[函数类型] 函数名 (函数参数表) /* 函
数说明部分 */

 { 说明语句部分;
 执行语句部分;
部分 */ /* 函数体
 }

下例的函数体部分为空，是一个空函数，什
么也不做，
但是是合法的。

```
dump()  
{  
}
```





5、语句 statement

如： `c=a*b; i=i+1;`

每条语句（包括数据定义）的分号必不可少。
语句书写格式自由，一行可写几句，一句可写在多行。

C 语言语句区分大小写。





2.2.2 头文件声明

程序的头文件声明由预“文件包含”预处理命令

实现，一般形式为：

#include <头文件名.h> 形式一

#include “头文件名.h” 形式二

注意：系统定义的头文件尽可能用形式一
用户自定义的头文件可用形式二

注意：在 TC 中对于 scanf 和 printf 函数
#include <stdio.h> 可省略，但其他函数不可。

在别的编译器中 #include <stdio.h> 不能省略。

系统的头文件在 \tc\include 目录下



2.2.3 注释

可用 `/**/` 来表示注释部分。 `/**/` 必须成对出现。

注释为了增加程序的可读性，与程序代码无关。

如果想暂时不执行末一段程序语句，
又不想删除。用 `/* */` 将这段程序括起来。





2.3、一个有趣的例子

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

```
main() {
```

```
    printf("          \\\\ \ - - //          \n");
```

```
    printf("          ( @ @ )          \n");
```

```
    printf(" ┌──────────────────o00o-( )-o00o──┐          \n");
```

```
    printf(" | 大家好;                                |          \n");
```

```
    printf(" | C 语言的学习现在开始了。                |          \n");
```

```
    printf(" |                                0ooo          |          \n");
```

```
    printf(" ┌──────────────────ooo0──-( )──┐          \n");
```

```
    printf("          ( ) ) /          \n");
```

```
    printf("          \\\ ( ( /          \n");
```



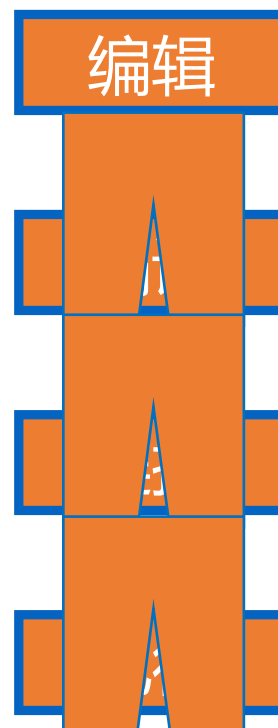
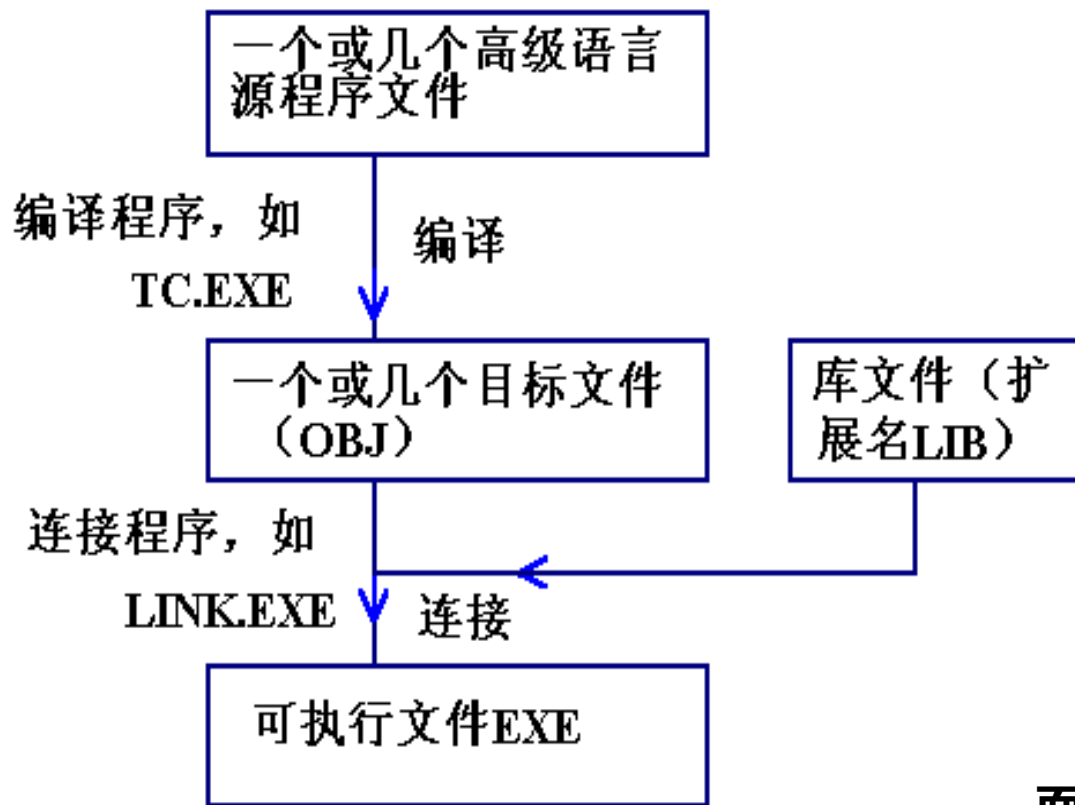
```
printf("*/. . . * . \n");
printf(" . \\* . [] * _ \n");
printf(" */ . . /\\ ~~~~~, \\ . | ◆ \n");
printf(" \\* , /, .., \\ , ..... , \\ . ◆ \n");
printf(" || ■ # ■ 田 田 ■ | ■ ◆ \n");
printf(" || ■ ■ ■ ' |' ■ o \n");
printf(" || ##■■■■■■■■■■ = ! \n");
}
```



三、程序编译原理简介



3.1 程序设计的步骤

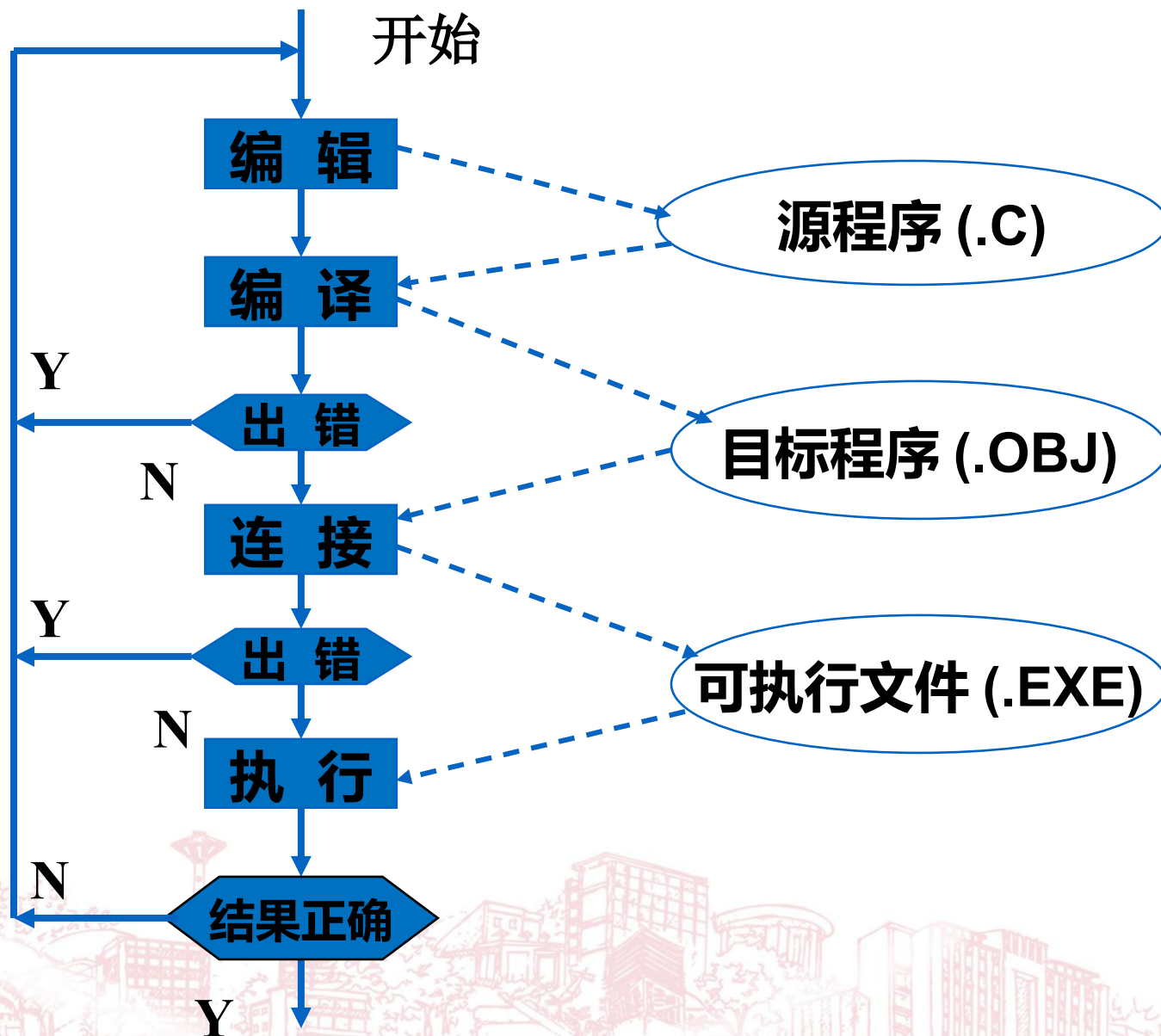


面粉和水
制

面包

烤

	源程序	目标程序	可执行程序
内容	程序设计语言	机器语言	机器语言
可执行	不可以	不可以	可以
文件名后缀	.c	.obj	.exe





- 1、输入源程序，保存成 *.c，如 excisel.c
- 2、编译 compile: 对源代码（文本文件）查错，
若
有错，提示错误信息。若没有错误生成同名的目标文件，如 excisel.obj
- 3、链接 link: 将构成整个程序的多个目标文件（目前我们的程序多由一个文件构成），以及使用到的系统函数所在的库函数连接成一个可执行文件，可执行文件的后缀一般为 excisel.exe.
- 4、运行 F: > excisel





C语言的特点：

C语言简洁、紧凑，书写形式自由。

C语言语法简洁，运算符和库函数丰富

C语言具有丰富的数据类型

C语言具有低级语言的功能

C语言程序移植性好





1、语言简洁、紧凑，使用方便灵活，程序书写形式自由。

有 32 个关键词（具特殊含义的词，如类型定义、控制语言等）。根据关键字的作用，可分其为数据类型关键字、控制语句关键字、存储类型关键字和其它关键字四类。

1）数据类型关键字（12 个）：char, double, enum, float, int, long, short, signed, struct, union, unsigned, void

2）控制语句关键字（12 个）：break, case, continue, default, do, else, for, goto, if, return, switch, while

3）存储类型关键字（4 个）：auto, extern, register, static

4）其它关键字（4 个）：const, sizeof, typedef, volatile



算术运算符 : + - * / % ++ --
关系运算符 : < <= == > >= !=
逻辑运算符 : ! && ||
位运算符 : << >> ~ | ^ &
赋值运算符 : = 及其扩展
条件运算符 : ? :
逗号运算符 : ,
指针运算符 : * &
求字节数 : sizeof
强制类型转换 : (类型)
分量运算符 : . ->
下标运算符 : []
其它 : () -



3、丰富的数据类型

有整、实、字符、枚举、数组、指针、结构、共用等数据类型，还能对复杂的数据结构（表、树、堆栈）进行操作。并为用户提供了自定义的数据类型。





4、具有高、低级语言的功能
直接访问物理地址和接口地址，能
进行位操作。
能进行科学计算、数据处理、绘图
、CAI、人工
智能等方面的应用。

5、可移植性好
程序不作或稍作修改就可在其它机
器上运行。





03

PART THREE

编写简单的 C 语言程序





初识 scanf 函数：

输入数据使用输入函数：

```
scanf("%d%d",&a,&b);
```

说明：

1. 输入整型数 (int) 用 % d , 输入实数类型 (float) 用 % f , 输入几个变量值就用几个格式符；例如：

```
scanf("%f%f%f",&a,&b,&c);
```

2. 第二部分参数是变量的名称前加运算符 “ & ”





初识 printf 函数

简单输出数据：

1. 只输出简单信息：

```
printf("Hello students ! \n");
```

2. 输出变量值：

```
printf("sum is %d\n", sum);
```





程序体的基本语法要素

【总结 1】

- 由若干程序语句按顺序书写，按顺序执行；
- 每条语句以分号 “ ; ” 结束；

一个复杂的语句也可以写成几行，用反斜杠 \ 表示续行；几个简单语句也可写在一起，都必须用 “ ; ” 作为每句结束标志。

- 通常变量定义放在开始部分；
- 语句后面可以有注释，注释内容放在 /* */ 内





C 语言标识符定义规则

1. 标识符只能由字母、数字和下划线三种字符组成，且第一个字符必须为字母或下划线。

例如： `sum,average,_total,Class,day,stu_name,p405`

不合法的标识符： `M.D.John,$123,#33,3D64,a>b`

2. 区别大小写。

例如： `sum` 不同 `Sum` 。 `BOOK` 不同 `book` 。

3. 标识符长度有限制 (TC 最多 32 个字符，有的系统最多 8 个)。

例如： `student_name,student_number` 如果取 8 个，这两个标识符是相同的。

4. 标识符不能与“关键词”同名，见教材 13 页。

5. 建议：标识符命名应当有一定的意义，做到见名知义。



内容总结及回顾

1. C 程序的基本结构：函数； main 框架；

预处理命令在前；有且只有一个 main 函数、函数组成（函数说明部分、函数体）；函数体内顺序：先变量定义，再其它语句。

2. 简单变量定义： int 、 float

3. 标识符的命名规则。

4. 简单使用 scanf 输入数据、 printf 输出数据





- C 语言字符集（略）

- 标识符：变量名、符号常量名、函数名、数组名、类型名等。

- 关键字（自学）

- C 语言语句及表达式：

变量定义语句、赋值语句、函数调用语句、空语句、表达式

- 运算符：算术运算符（除法）、赋值运算符、

sizeof 运算符、关系运算符。

优先级见附件，括号优先、括号一律用小括号、多层括号由内向外算；

- 分隔符：空格（源程序中起到分隔元素的作用）

空格、回车、**Tab**（输入数据的分隔）

- 其它符号：略



04

PART FOUR

算法





1、算法的概念及特性

- (1) 有穷性：
- (2) 确定性：
- (3) 有 0 个或多个输入：
- (4) 有 1 个或多个输出：
- (5) 有效性：





2、算法的表示方法

- 自然语言
- 传统的程序流程图
- N-S 流程图
- 伪代码或计算机语言描述





结构化程序设计的 三种基本结构

顺序结构

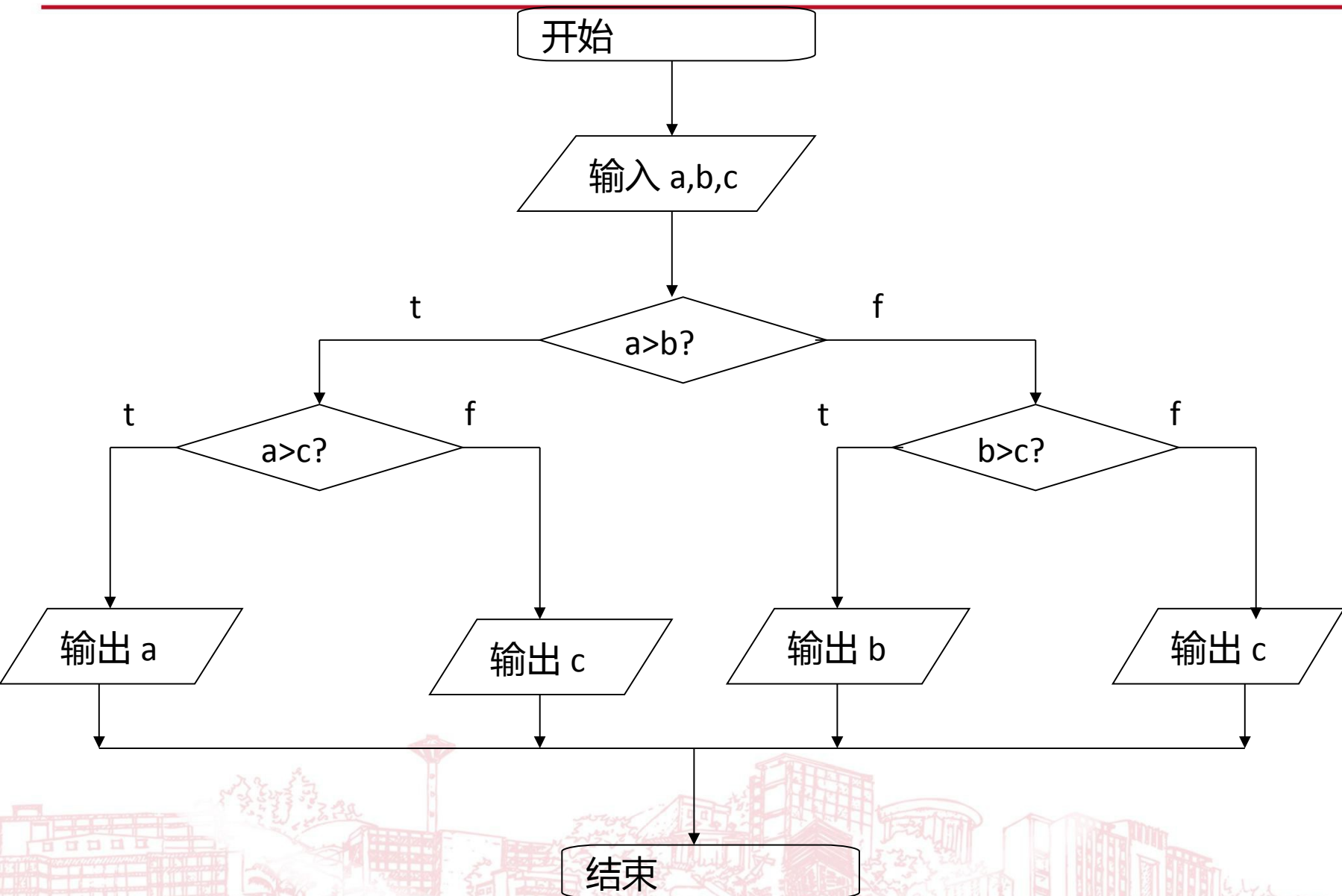
选择结构

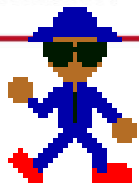
循环结构





传统的程序流程图例图





N-S 图

即结构化程序设计流程图，提供了描述三种基本逻辑结构的图形工具，与传统的程序流程图对照描述如下：

1、顺序结构

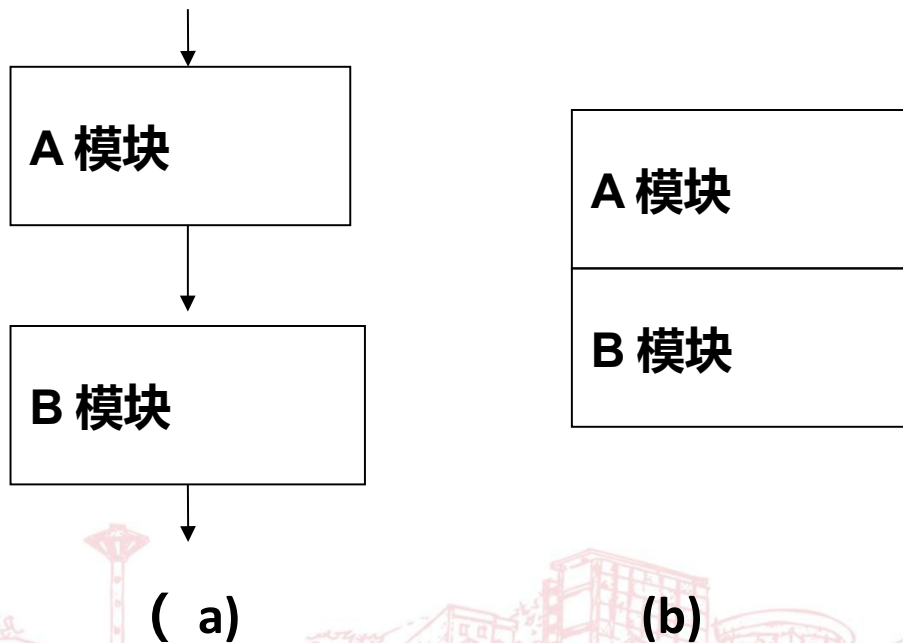
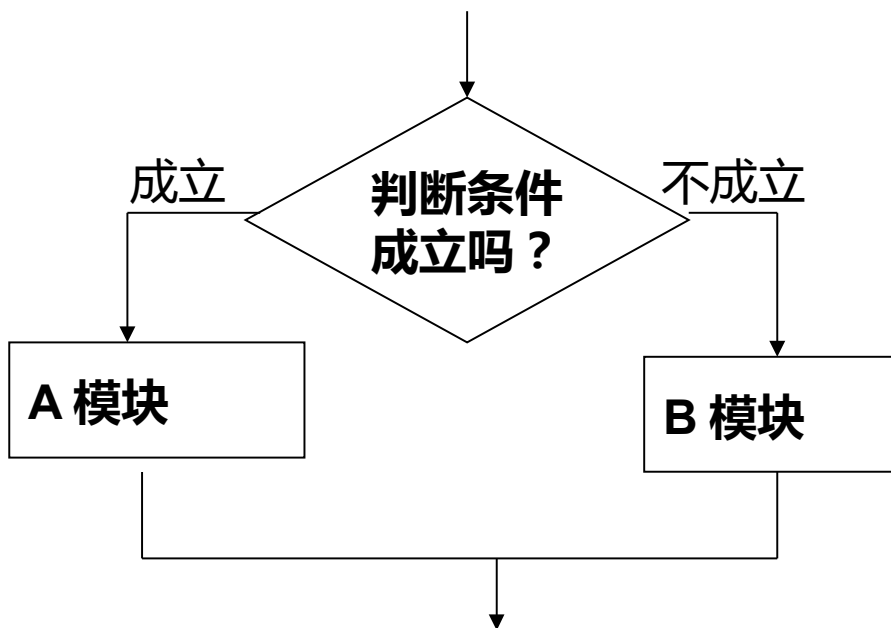


图 1 顺序结构

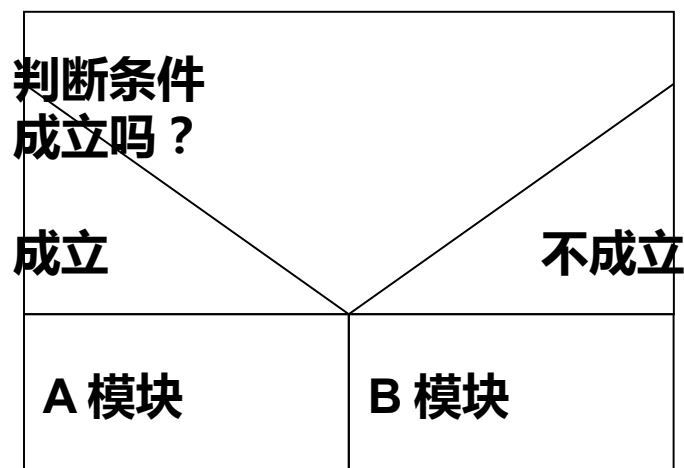


N-S 图

2、选择结构（分支结构）



(a) 流程图表示

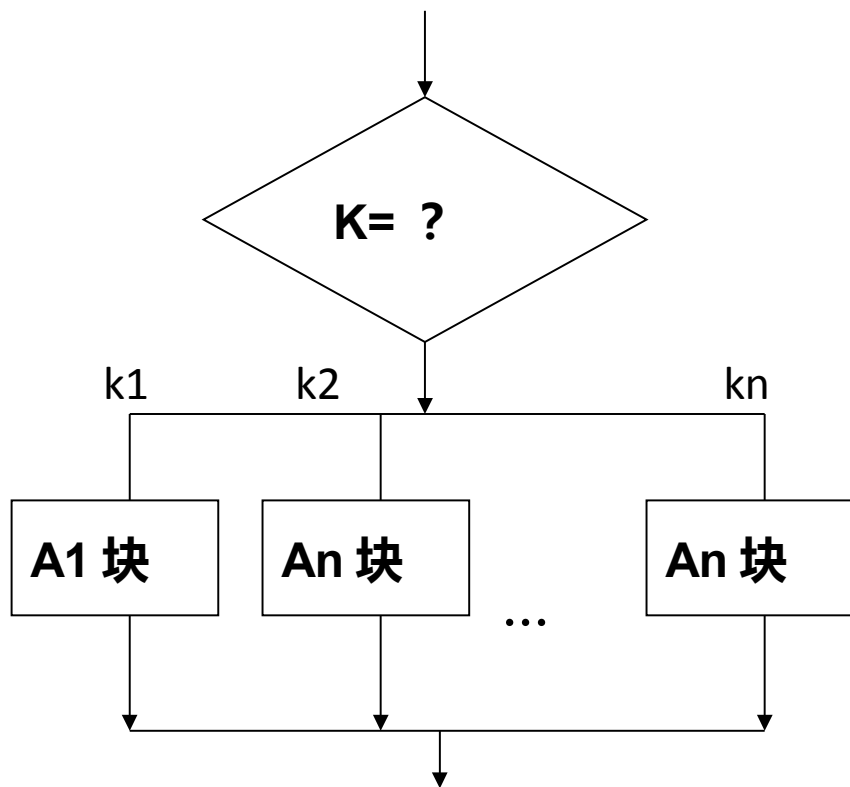


(b)N-S 图表示

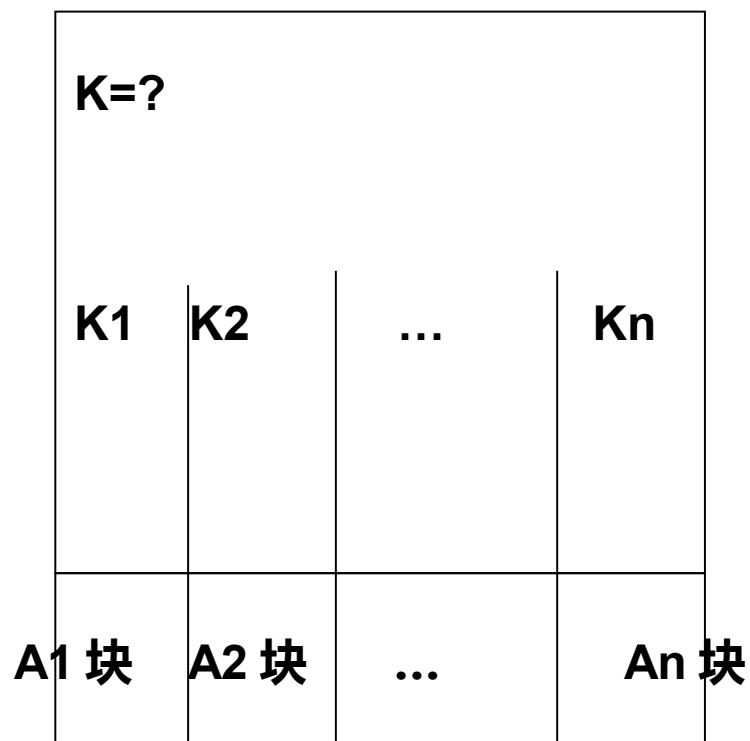
图 1.4 选择结构



由分支结构派生出来的多分支结构



(a) 流程图表示



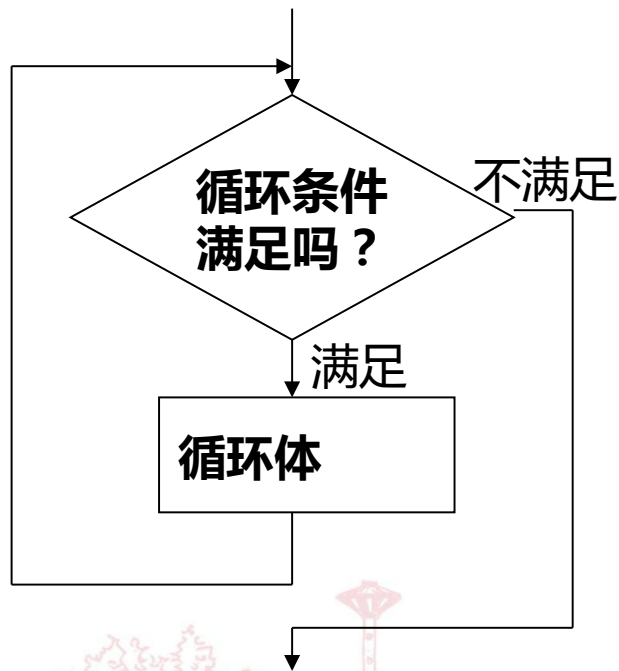
(b) N-S 图表

图 1.5 多分支结构

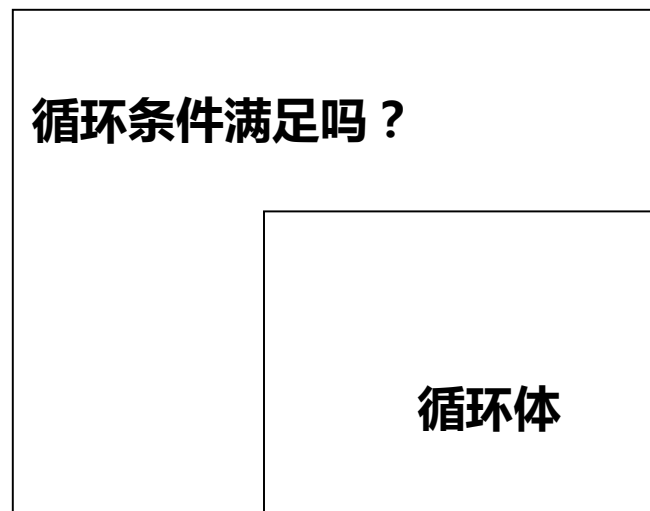


3、循环结构

a. 当型循环结构：先判断循环条件是否满足，当条件满足，执行一遍循环体，再判断循环条件，仍满足再执行一遍循环体……直到不满足条件，则退出循环，执行循环体后的下一语句。若第一次判断条件就不满足，则一次也不执行循环体。



(a) 流程图表示

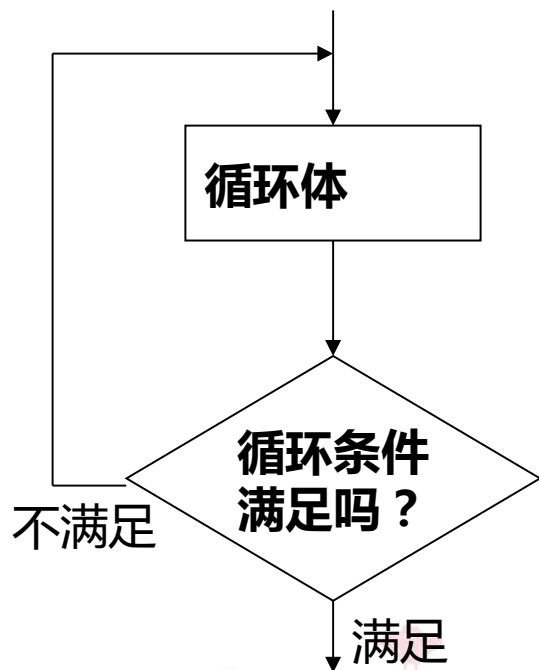


(b) N-S 图表示

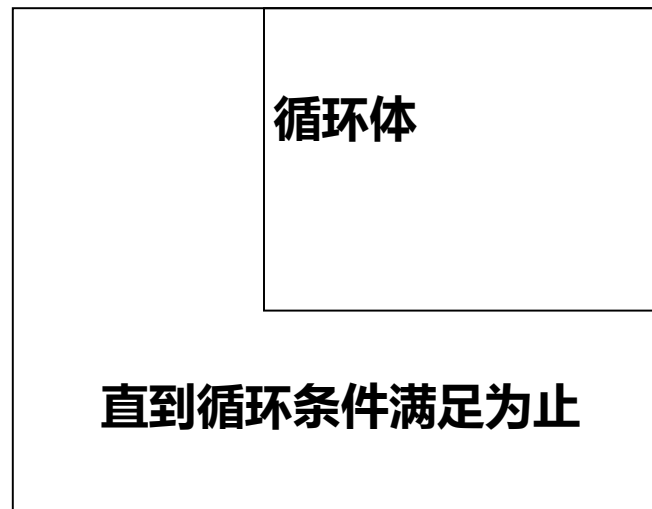
图 1.6 当型循环结构

N-S 图

b. 直到型循环结构：先执行一遍循环体，再判断循环条件是否满足，如不满足再执行一遍循环体……直到条件满足，则退出循环，执行循环体后的下一语句。这种循环结构至少要执行一遍循环体。



(a) 流程图表示



(b) N-S 图表示

图 1.7 直到型循环结构



SOC 达成度的总体评价

效果

教学反思与小结

效果





广东岭南职业技术学院
GUANGDONG LINGNAN
POLYTECHNIC

感谢各位聆听

Thanks for Listening



现代制造学院 | 香港铸业学院

模具设计与制造专业

