



福州大学至诚学院
FUZHOU UNIVERSITY ZHICHENG COLLEGE

高级语言程序设计 (C语言与数据结构)

杨雄

83789047@qq.com



第一章	C语言的发展、特点与程序结构
第二章	基本数据类型、运算符和表达式
第三章	基本语句与顺序结构
第四章	选择结构
第五章	循环结构
第六章	数组
第七章	函数
第八章	指针
第九章	结构体



推荐参考书：

谭浩强主编 《C 程序设计》（第二版）

北京：清华大学出版社



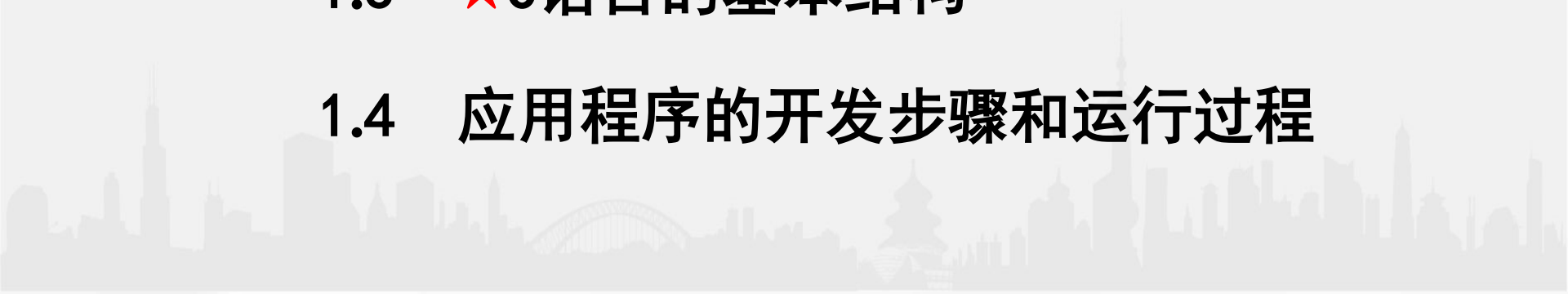
第一章 C语言的发展、特点与程序开发基本知识

1.1 C语言的发展与特点

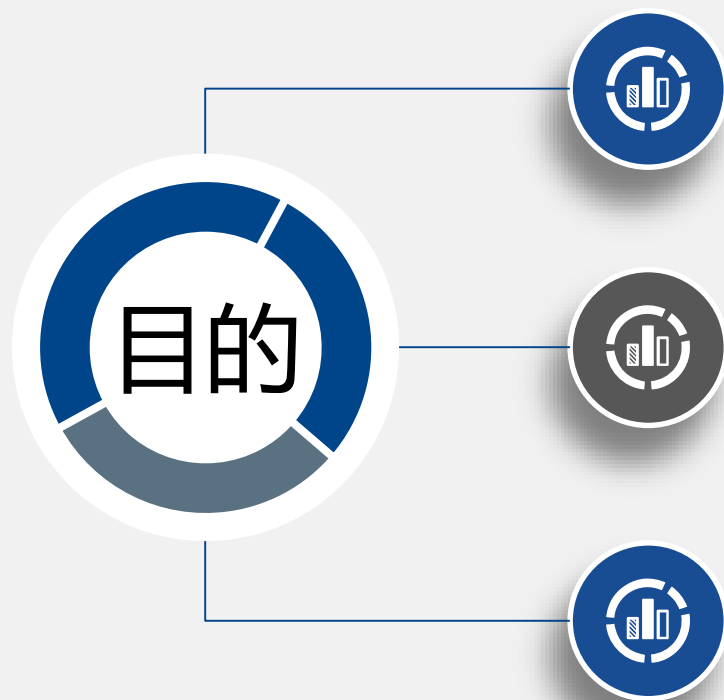
1.2 程序设计的基本方法

1.3 ★C语言的基本结构

1.4 应用程序的开发步骤和运行过程



学习目的



了解 C 语言的背景

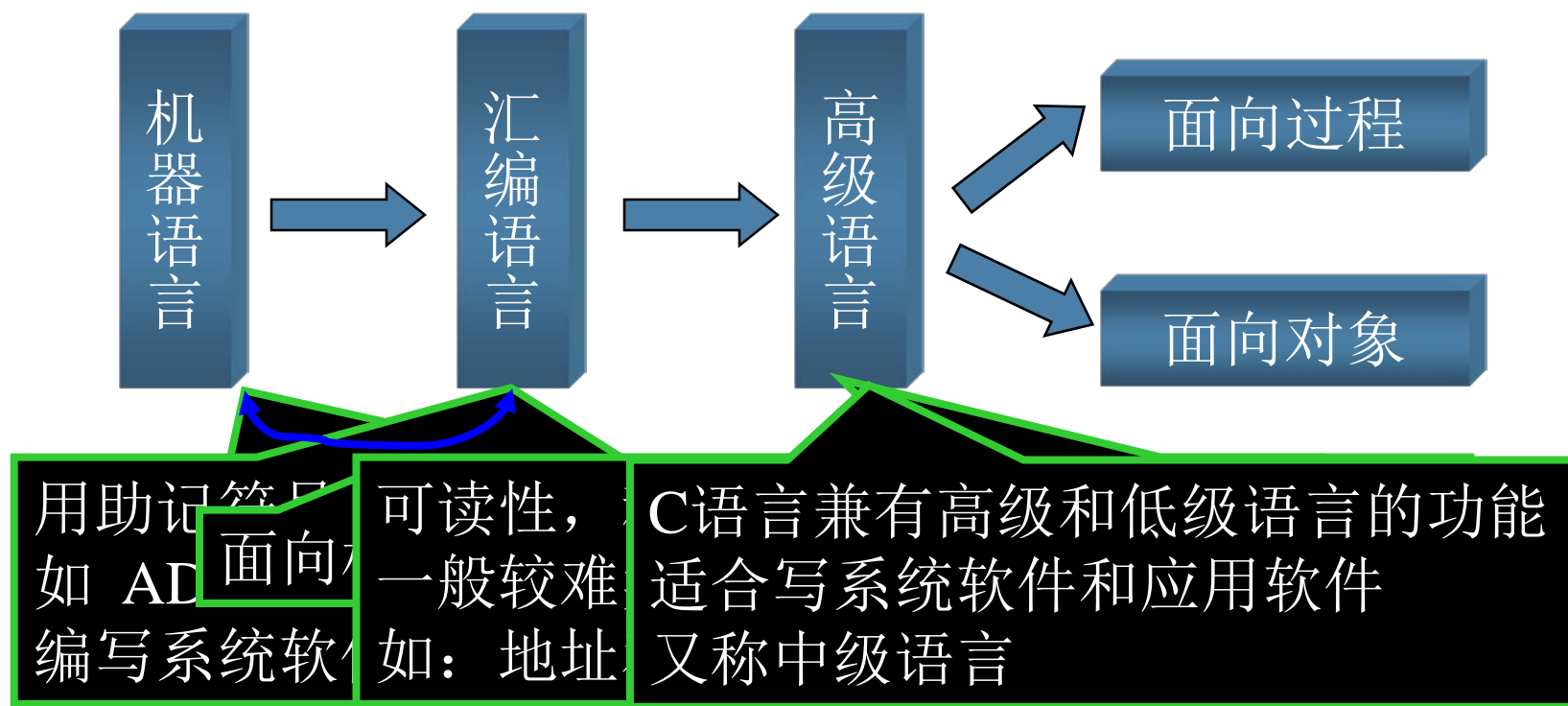
掌握 C 语言程序的结构

理解 C 语言程序设计的风格

Part.1

1.1 C语言的发展与特点

1.1 C语言的发展和特点



1.1 C语言的发展和特点

➤ C语言的发展

- C语言是国际上广泛流行的高级语言。
- C语言是在B语言的基础上发展起来的。
- 1973年贝尔实验室的D. M. Ritchie 在B语言的基础上设计出了C语言，对B取长补短，并用之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。

1.1 C语言的发展和特点

1、**C89标准**，ANSI制定的第一个C语言标准，在1989年被正式采用，故称C89，也称**ANSI C**。

该标准随后ISO采纳，成为国际标准(ISO/IEC 9899:1990)。

目前流行的C语言编译系统大多是以ANSI C为基础进行开发的。

2、**C99标准**，由ISO制定于1999年(ISO/IEC 9899:1999)，故称为C99。

3、**C11标准**，由ISO制定于2011年(ISO/IEC 9899:2011)，故称为C11。

1.1 C语言的发展和特点

➤ C语言的特点

- (1) 语言简洁、紧凑,使用方便、灵活。 32个关键字、9种控制语句,程序形式自由
- (2) 运算符丰富。34种运算符
- (3) 数据类型丰富,具有现代语言的各种数据结构。
- (4) 具有结构化的控制语句 , 是完全模块化和结构化的语言。
- (5) 语法限制不太严格,程序设计自由度大。

1.1 C语言的发展和特点

➤ C语言的特点

- (6) 允许直接访问物理地址, 能进行位操作, 能实现汇编语言的大部分功能, 可直接对硬件进行操作。兼有高级和低级语言的特点。
- (7) 目标代码质量高, 程序执行效率高。只比汇编程序生成的目标代码效率低10%-20%。
- (8) 程序可移植性好(与汇编语言比)。基本上不做修改就能用于各种型号的计算机和各种操作系统。

1.1 C语言的发展和特点

➤ C语言的特点

问题：既然有了面向对象的C++语言，为什么还要学习C语言？

解释1：C++是由于开发大型应用软件的需要而产生的，并不是所有的人都要去编写大型软件；

解释2：面向对象的基础是面向过程。C++是面向对象的语言，C是面向过程的，学起来比C语言困难得多，所以不太适合程序设计的初学者。

Part.2

1.2 程序设计的基本方法

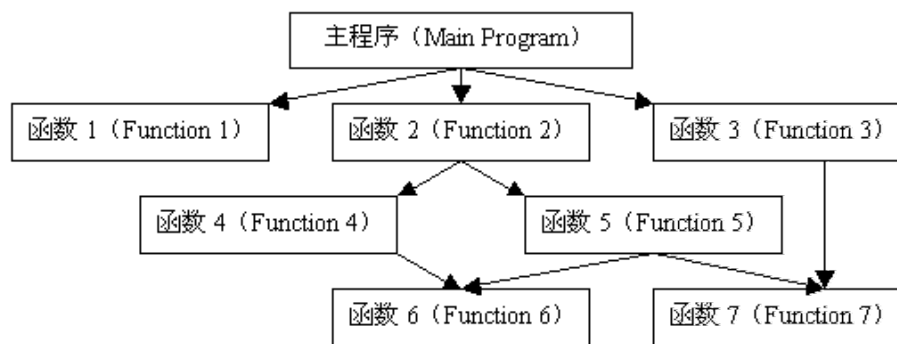
1.2 程序设计的基本方法

➤ 面向过程的程序设计

- 所谓面向过程的程序设计，是指利用面向过程的语言工具 (如 Pascal、Fortran和C语言等)进行程序开发的各项活动。

基本思想：

把一个需要求解的复杂问题分为若干个模块来处理，每个模块处理一个子问题；设计时遵循自顶向下、逐步细化、模块化设计和结构化编码的原则。



优点： 编程简单、结构性强、可读性好；
遵循这种结构的程序只有一个入口和一个出口。

缺点： 存在数据与程序模块的分离和程序的可重用性差等问题。

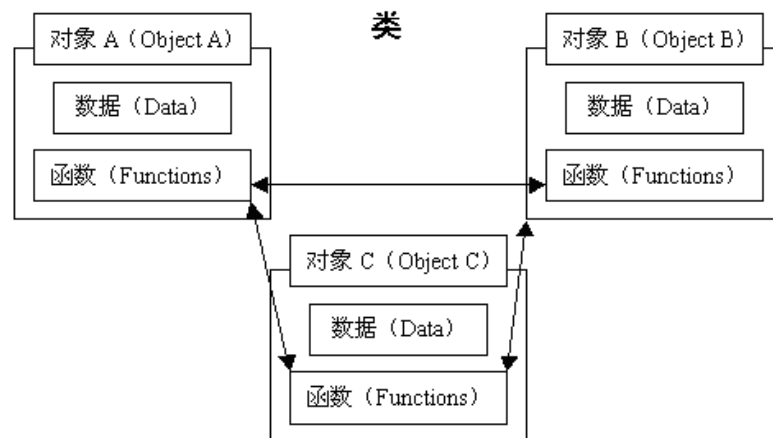
1.2 程序设计的基本方法

➤ 面向对象的程序设计

- 面向对象的程序设计将一些新的理念和结构化程序设计中好的思想相融合，提供了一种全新的程序设计方法。

基本思想

将一个需要解决的问题分解为一系列实体（对象），然后围绕这些对象建立数据和函数；函数的功能决定了该对象的行为。规定一个对象的函数可以访问另一对象的函数，但一个对象的数据不能应用于另一对象的函数中。



几个基本概念：

类、对象、属性、事件、方法

三类机制：

封装、继承、多态

Part.3

1.3 C语言的基本结构

1.3 C语言的基本结构

C语言是结构化程序设计语言，有三种基本结构。

- ✧ 顺序结构

- ✧ 选择结构

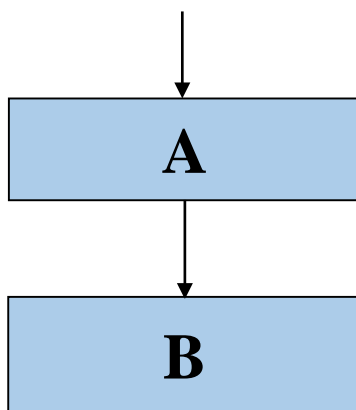
- ✧ 循环结构

实践证明, 任何复杂的算法都可以用这三种基本结构来描述。

1.3 C语言的基本结构

➤ 顺序结构

矩形框 — 用于表示顺序结构。执行时按语句的先后顺序执行。先执行语句A，再执行语句B。

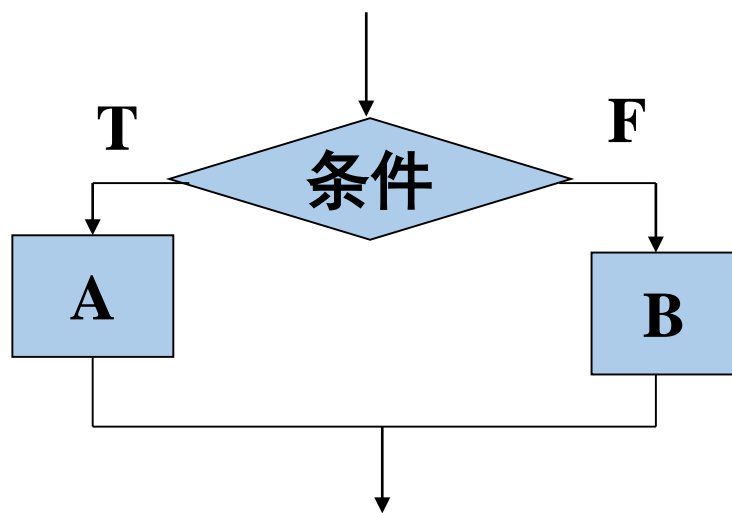


顺序结构的流程图

1.3 C语言的基本结构

➤ 选择结构（分支结构）

根据条件的真或假选择执行不同的操作内容。当条件成立执行语句A，否则执行语句B。



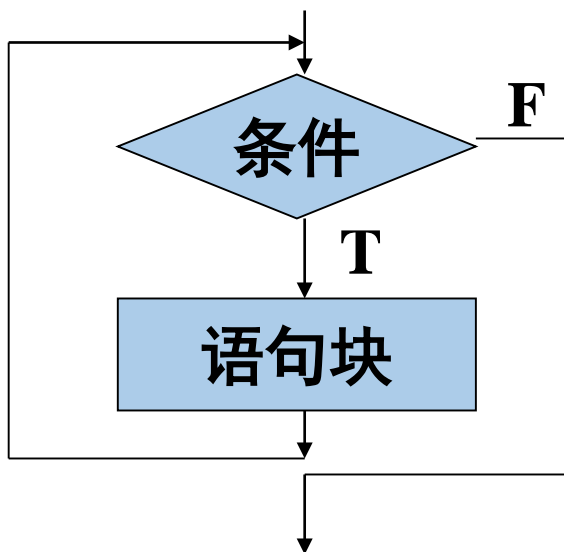
选择结构的流程图

1.3 C语言的基本结构

➤ 循环结构

(1) 当型（while 型循环）

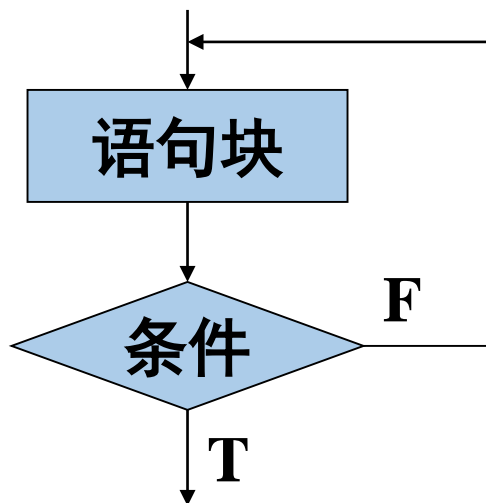
当条件成立, 执行循环体, 否则跳出循环体。



1.3 C语言的基本结构

➤ 循环结构

首先执行循环体，若条件不成立继续执行循环体，直到条件成立为止。



1.3 C语言的基本结构

```
#include <stdio.h>
void main( )
{
    printf ("This is a C program.\n")
}
```

```
#include <stdio.h>
void main( )
{
    printf ("This is a C program.\n');
}
```

说明： 本程序的作用是输出一行信息：

This is a C program.

#include <stdio.h>

/*文件包含*/

void main()

/*主函数 */

{

/*函数体开始*/

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

/*输出语句*/

}

/*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>

说明： 输出一行信息：sum is 579

例1.2 求两数之和

```
#include <stdio.h>
void main( )      /*求两数之和*/
{
    int a, b, sum;  /*声明，定义变量为整型*/
    /*以下3行为C语句 */
    a=123; b=456;
    sum=a+b;
    printf(" sum is %d\n" , sum);
}
```

说明： `/*……*/`表示注释。注释只是给人看的，对编译和运行不起作用。所以可以用汉字或英文字符表示，可以出现在一行中的最右侧，也可以单独成为一行。

例1.3 求3个数中较大者

```
#include <stdio.h>
void main( )    /* 主函数 */
{
```

```
    int max(int x, int y); /* 对被调用函数max的声明 */
```

```
    int a, b, c;          /* 定义变量a、b、c */
```

```
    scanf(" %d, %d", &a, &b); /* 输入变量a和b的值 */
```

```
    c = max(a, b); /* 调用max函数, 将得到的值赋给c */
```

```
    printf(" max=%d\\n", c);
```

• 程序运行情况如下:

- 8, 5 ↙ (输入8和5赋给a和b)
- max=8 (输出c的值)

说明: 本程序包括main和被调用函数max两个函数。max函数的作用是将x和y中较大者的值赋给变量z。return语句将z的值返回给主调函数main。

```
int max(int x, int y);
{
    int z;
    if (x > y) z = x;
    else z = y;
    return (z);
}
```

1.3 C语言的基本结构

C程序:

- (1) C程序总是从`main`函数开始执行的, 与`main`函数的位置无关, `main()`可以置于程序的任何位置。
- (2) 程序中需有预处理命令(如 `#include <stdio.h>`), 预处理命令通常放在程序的最前面。
- (3) **分号**是C语句结束的标志, 每个语句和数据定义后必须有一个分号。预处理命令、函数头和花括号“`{}`”之后不能加分号。

1.3 C语言的基本结构

C程序：

- (4) C语言本身没有输入输出语句。输入和输出的操作是由库函数scanf和printf等函数来完成的。C对输入输出实行“函数化”。
- (5) C程序**严格区分大小写**。一般变量、语句等用小写字母书写；符号常量等用大写字母书写。

1.3 C语言的基本结构

函数：

- C语言的函数有**两种**：**标准库函数**和**用户定义的函数**。
 - 用户自定义函数，可有可无，数目不限。
如例1.3中的 `int max(int x , int y)`
 - C语言提供的库函数，如输出函数**`printf()`**和输入函数 **`scanf()`**。

1.3 C语言的基本结构

C程序：

- (1) C程序是由**函数**构成的。这使得程序容易实现模块化。
- (2) 一个函数由**两部分组成**：

函数头：**说明函数的类型、名字、参数及参数的类型**

例1.3中的max函数首部

```
int max(int x, int y )
```

函数体：{} 内的部分。若一个函数有多个 {}, 则最外层的一对 {} 为函数体的范围。

函数体包括两部分：

声明部分： `int a, b, c;` 可缺省

执行部分：由若干个语句组成。 可缺省

1.3 C语言的基本结构

注意:

函数的声明部分和执行部分都可缺省，例如：

```
void dump ( )  
  
{  
  
}
```

这是一个空函数，什么也不做，但是合法的函数。

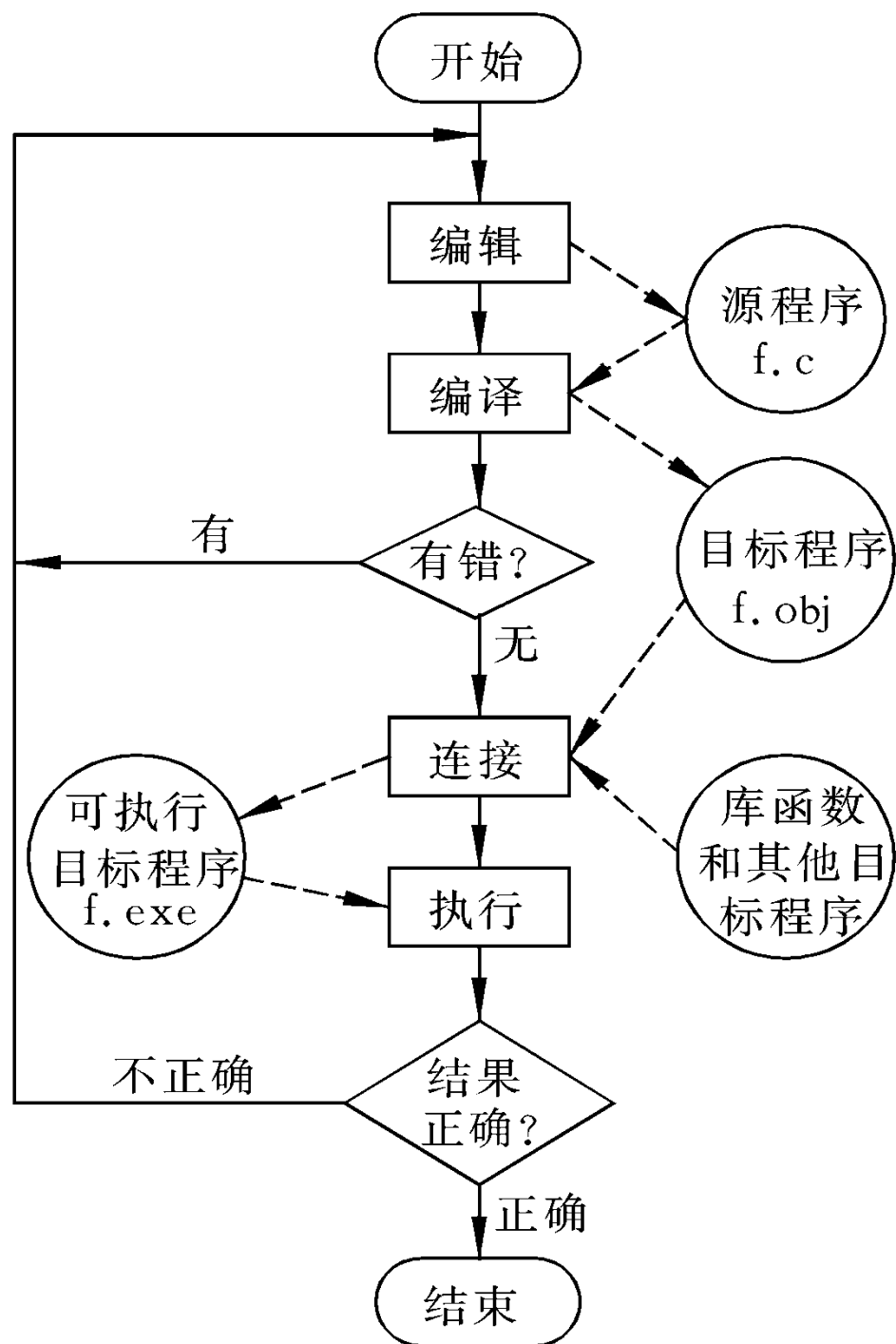
Part.4

1.4 应用程序的开发步骤和运行过程

1.4 应用程序的开

一、运行C程序的步骤

- 上机输入与编辑源程序
- 对源程序进行编译
- 与库函数连接
- 运行目标程序



习 题

1. 一个C语言程序是由_____组成

- a) 主程序 b) 子程序 c) 函数 d) 过程

2. 一个完整的C语言的程序

- ✓ a) 至少包含一个主函数
b) 至少包含一个主函数与一个子函数
c) 必须由一个主函数与一个以上的子函数组成
d) 必须由一个主函数与一个子函数组成

习 题

3. C语言程序的执行总是从_____。

- ✓ a) main函数的入口开始
- b) 编译预处理命令开始
- c) 源程序的第一个函数开始
- d) 源程序的第一行语句开始

习 题

4.在C语言的源程序中，主函数的位置 ____。

- a) 必须在源程序的最后
- ✓ b) 任意
- c) 必须在源程序的开头

5.一个函数一般由两部分组成,它们分别是函数头和函数体。

6.函数体使用 { 符号开始，使用 } 符号结束。

习 题

7. C程序的每条语句的结束标志是;符号。
8. 在C程序中, 注释以/*符号开始, 使用*/符号结束。
9. 结构化程序的三种基本结构是 顺序结构、选择结构、循环结构。

习题

10. 判断下述说法是否正确。如不正确, 说明理由。

- a) **printf** 函数总是从新行的起始位置开始打印。 ✗
- b) 计算机会把 `/*`和`*/`之间的注释文本打印到屏幕上。 ✗
- c) 所有的变量在使用前都必须予以声明。 ✓
- d) C语言认为变量 **number** 和 **NuMbEr** 是相同的。 ✗
- e) 声明语句可放在函数体中的任何位置。 ✗
- f) 打印三行输出的C语言程序必须用三条 **printf** 语句。 ✗
- g) **printf** 函数的格式控制串中使用转义字符 `\n` 会把光标定位到屏幕上下一行的开始位置。 ✓

Part.4

总结

总 结

1. C语言是面向过程的高级程序语言。
2. C语言程序三种基本结构：顺序结构、选择结构和循环结构。
3. C语言程序C总是从main函数开始执行的, 位置无关；
分号是C语句结束的标志
4. C程序是由函数构成的，函数是由函数头和函数体构成。函数分为库函数和用户自定义函数。
5. C 程序的开发步骤：编辑、编译、链接和运行