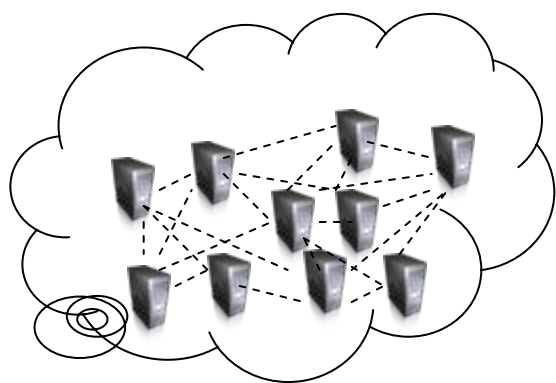


# 厦门大学非计算机专业本科生公共课 (2011-2012第2学期)



## C语言程序设计

林子雨

厦门大学计算机科学系

E-mail: [ziyulin@xmu.edu.cn](mailto:ziyulin@xmu.edu.cn)

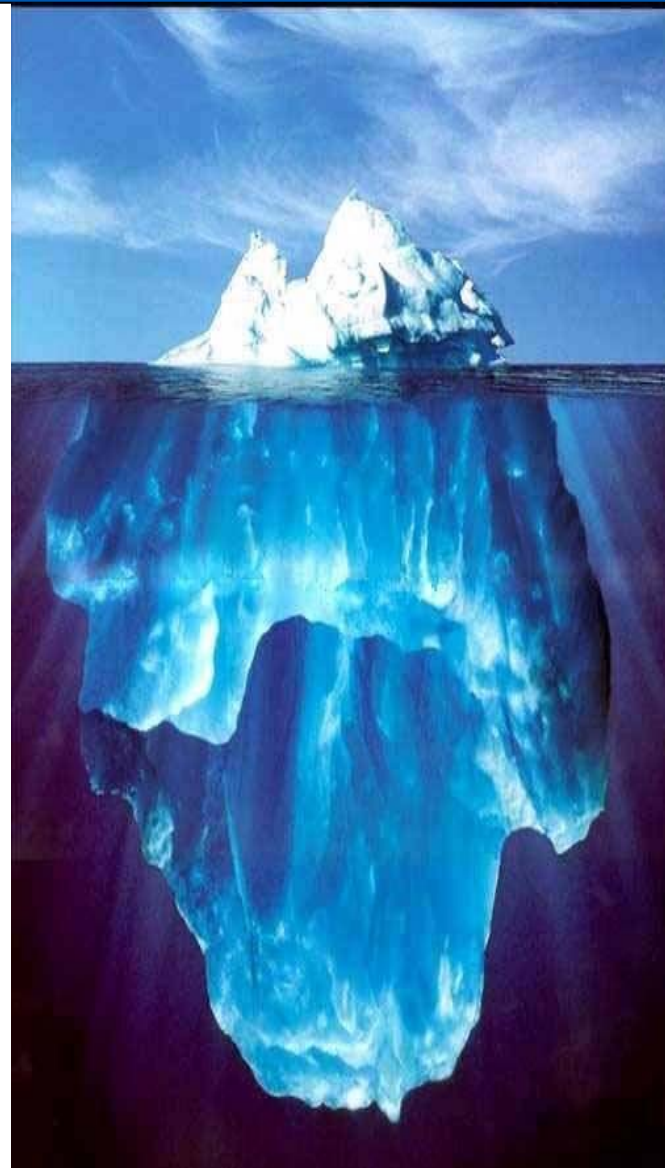
个人主页: <http://www.cs.xmu.edu.cn/linziyu> ▶▶





# 课程提要

- 第一章 绪论
- 第二章 C语言基础
- **第三章 结构化程序设计**
- 第四章 选择结构
- 第五章 循环结构程序设计
- 第六章 函数
- 第七章 编译预处理
- 第八章 数组
- 第九章 结构体、共用体和枚举类型
- 第十章 指针





# 第3章 结构化程序设计

- 3.1 结构化程序设计方法
- 3.2 语句的概念
- 3.3 程序的三种基本结构
- 3.4 赋值语句
- 3.5 输入输出函数
- 3.6 顺序结构程序设计举例





## 3.1 结构化程序设计方法

- 结构化程序设计方法是一整套指导软件开发的方法，涵盖了系统分析、系统设计和程序设计这三个方面的内容。
- 采用自顶向下、逐步细化、模块化的方法进行程序设计
- 强调程序设计风格和程序结构的规范化，提倡清晰的程序结构
- 基本思路是：把一个复杂问题的求解过程分阶段进行，每个阶段处理的问题都控制在人们容易理解和处理的范围内。
- 具体实现步骤：
  - (1) 按自顶向下逐步细化的方法对问题进行分析和设计；
  - (2) 系统的模块设计；
  - (3) 结构化编码。





## 3.2 语句的概念

- 3.2.1 表达式语句
- 3.2.2 控制语句
- 3.2.3 复合语句
- 3.2.4 空语句





## 3.2.1 表达式语句

- 在C语言中，只要在任何表达式的结尾加上分号“;”，就构成了表达式语句，如：

```
i=1,j=2,k=3;
```

```
j+k; (合法但无效)
```

```
i++;
```

```
fun(j,j+k,6);
```





## 3.2.2 控制语句

- 控制语句是程序中用来控制执行次序的语句，它能够影响其他语句的执行次序。
- C语言提供了9种控制语句，分为3类：
  - (1) 选择语句: `if...else...`, `switch`
  - (2) 循环语句: `while`, `do...while`, `for`
  - (3) 转向语句: `break`, `continue`, `goto`, `return`





## 3.2.3 复合语句

- 复合语句是由大括号{}把一些语句括起来组成。  
例如：{temp=x;x=y;y=temp;}
- 复合语句在语法上相当于一个语句
- 复合语句可以嵌套
- 复合语句在编程中主要用于两种情形：
  - (1) 语法要求用一条语句，但又难以用一条简单语句表达清楚的情形，如作为for、while语句等的内嵌语句
  - (2) 形成局部化的封装体。如块语句中定义的变量只在本块范围内可用。
- 注意：复合语句内的各条语句都必须以分号“;”结尾，但在复合语句结束标志右大括号}后面则不能加分号。







## 3.2.4 空语句

- 没有表达式的语句称为**空语句**，即它是只有一个分号“;”的语句；
- 空语句也可看成是一个特殊的表达式语句，但不做任何事情
- 作用是：用于语法上需要一条语句的地方，而该地方不需要做任何事情

例如：`for(i=1;i<=100000;i++);`用于“耗时”





## 3.3 程序的三种基本结构

1966年，计算机科学家经过研究证实，所有程序，不论简单还是复杂，都能够由三种控制结构组合来实现：

- 顺序结构
  - 物理上相邻的多条语句在执行时，按照其物理排列的前后顺序，一条一条地顺序执行。
- 选择结构
  - 流程不按照语句在程序中出现的先后顺序逐条执行，而是根据判断项的值有条件地选择部分语句执行。
- 循环结构
  - 根据需要反复执行程序中的某些语句





## 3.4 赋值语句

- 格式：变量=表达式;
- 例3.4.1 交换变量x和y的值。

```
#include <stdio.h>
main()
{
    int x=6,y=8,temp;
    printf("x=%d,y=%d\n",x,y);
    temp=x; //将x的初值赋予变量temp
    x=y; //仅改变变量x的值，y的值不变
    y=temp; //变量y被赋予新的值，原值被覆盖
    printf("x=%d,y=%d\n",x,y);
}
```





## 3.4 赋值语句

例3.4.2 输入一个4位整数，然后打印出它的4位数字的和

```
#include <stdio.h>
void main()
{
    int n,a,b,c,d,sum;
    printf("Input n:");
    scanf("%d",&n);
    a=n%10; //求个位数
    b=n/10%10; //求十位数
    c=n/100%10; //求百位数
    d=n/1000; //求千位数
    sum=a+b+c+d;
    printf("n=%d,sum=%d\n",n,sum);
}
```





## 3.5 输入输出函数

- 3.5.1 格式输出函数printf
- 3.5.2 格式输入函数scanf
- 3.5.3 字符输出函数putchar()
- 3.5.4 字符输入函数getchar()





## 3.5.1 格式输出函数printf

- 3.5.1.0 printf概述
- 3.5.1.1 整型格式控制符
- 3.5.1.2 浮点型格式控制符
- 3.5.1.3 字符型格式控制符
- 3.5.1.4 字符串格式控制符





## 3.5.1.0 printf概述

- printf函数的调用形式为：  
`printf(“格式控制字符串”,表达式1,表达式 2,...,表达式n);`
- 当n=0时，调用形式：  
`printf(“格式控制字符串”);`
- 格式控制符：总是由“%”字符开始，并以一个类型描述符结束，中间是一些可选的附加说明项。完整格式如下：  
`%[- 0 m.n l/h]格式字符`





## 3.5.1.1 整型格式控制符

- 用于控制整型数据的输出格式
- (1) 十进制形式：以十进制形式输出整型数据：
  - %d或%md 用于基本整型
  - %ld或%mlld 用于长整型
  - %u或%mu 用于无符号基本整型
  - %lu或%mlu 用于无符号长整型
- (2) 八进制形式：以八进制形式输出整型数据（无前缀0）：
  - %o或%mo 用于基本整型
  - %lo或%mllo 用于长整型
- (3) 十六进制形式：以十六进制形式输出整型数据（无前缀0x）：
  - %x或%mx 用于基本整型
  - %lx或%mlx 用于长整型

注意：**m**表示输出的整型数据所占总宽度（即列数）。如果数据的位数小于**m**，则左端补空格，若大于**m**，则按实际位数输出。如果在格式控制符中没有用**m**来说明数据所占的宽度，则以输出数据的实际位数为准。如：`printf(“%3d,%4d”,a,b);`

若`int a=12,b=12345`,则输出结果为：

□12,12345







## 3.5.1.2 浮点型格式控制符

- 以小数形式输出浮点型数据，格式控制符为：  
`%m.nf`或`%f`  
其中，`m`表示输出数据所占的总宽度，包括小数点所占的一列，`n`表示小数点后面所占的位数。
- 注意：在C语言中，用于输出单精度浮点型数据和双精度浮点型数据的格式控制符是一样的。`double`型数据输出使用`%lf`和`%f`无区别。

### 例3.5.3 输出实数时指定小数位数

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

```
main()
```

```
{
```

```
    float f = 234.567;
```

```
    printf("%f\n",f);
```

```
    printf("%10.2f\n",f);
```

```
    printf("%-10.2f\n",f);
```

```
    printf("%.2f\n",f);
```

```
    printf("%10f\n",f);
```

```
}
```

运行结果：

234.567001（系统默认输出全部整数和6位小数）

□□□□234.57

234.57□□□□

234.57

234.567001（占10位，浮点数在内存中存储误差）





## 3.5.1.3 字符型格式控制符

- 用于说明字符型数据的输出格式。其格式控制符为：

`%mc`或`%c`

其中，`m`表示输出的宽度，在输出字符的左边将要补`m-1`个空格

例如：`char c='A';printf(“%5c”,c);`

输出结果为“□□□□A”

注意：一个整数，只要它的值在0-127范围内，也可以用字符形式输出，在输出前，系统会将该整数作为ASCII码转换成相应的字符；反之，一个字符数据也可以用整数形式输出。

例3.5.4 字符数据的输出

```
main()
```

```
{
```

```
    char c='B';
```

```
    int i=66;
```

```
    printf(“%c,%d\n”,c,c);
```

```
    printf(“%c,%d\n”,i,i);
```

```
}
```

运行结果：

```
B,66
```

```
B,66
```





## 3.5.1.4 字符串格式控制符

- 用来输出一个字符串。
- (1) %s, 按紧凑格式输出字符串

例如: `printf(“%s%s”, “China”, “□Beijing”);`

输出字符串: `China□Beijing` (定界符双引号不输出)

- (2) %ms
- (3) %-ms
- (4) %m.ns
- (5) %-m.ns





## 3.5.2 格式输入函数scanf

- 当程序调用**scanf**函数时，用户可以直接从键盘向内存中的变量输入不同类型的数据。其调用形式为：  
**scanf**(格式控制字符串，输入项地址列表)
- 格式化字符参见教材78页表3.5.1。
- 注意：在使用**scanf**时，务必保证格式控制字符串中以%开头格式控制符的个数，与输入项地址列表的项数相同，且类型匹配，否则将产生不可预测的效果。





## 3.5.2 格式输入函数scanf

- (1)附加格式说明符m可以指定数据宽度  
scanf(“%3d%3d”,&a,&b);输入: 123456(回车)  
系统自动把123截取给a, 456截取给b
- (2)输入long int型数据必须用%ld, 输入double型数据必须用%lf
- (3)附加格式符\*使对应的输入数据不赋给相应变量  
double a; int b; float c;  
scanf(“%f,%2d,%\*d,%5f”,&a,&b,&c);  
输入4.6,25,123,2.34567(回车)  
赋值后, a=0 (double型使用%f是错误的), b=25, c=2.345
- (4)注意逗号和空格的使用, 不要使用‘\n’。





## 3.5.2 格式输入函数scanf

- 例3.5.5 用scanf函数输入数据

main()

{

```
int a;float b;double c;
```

```
scanf(“%d%f%lf”,&a,&b,&c);
```

```
printf(“a=%d,b=%f,c=%f\n”,a,b,c);
```

}

运行时按以下方式输入a,b,c的值:

6□3.14□8.967 (回车)

运行结果:

a=6,b=3.140000,c=8.967000





## 3.5.3 字符输出函数putchar()

- 调用形式为： `putchar(c)`;  
其中， `c`可以是字符型常量、字符型变量或整型变量
- 功能：在屏幕当前光标位置处，显示 `c`所表示的一个字符。

### 例3.5.6用 `putchar` 函数输出单个字符。

```
#include <stdio.h>           /*包含I/O标准库函数的头文件*/
main()
{
    char a, b,c;              //定义字符型变量a,b,c
    a='O';b='K';c='!';       //对变量a,b,c进行赋值
    putchar(a); putchar(b); putchar(c); //输出OK!
    putchar('\n');          //输出换行
    putchar('\x41'); putchar('\102');
}
```

运行结果：

OK!

AB





## 3.5.4 字符输入函数getchar()

- 字符输入函数getchar()没有参数，一般形式为：  
getchar();
- 功能：接收从键盘输入的一个字符。

### 例3.5.7输入单个字符

```
#include <stdio. h>
main()
{ char c; int i;
  c=getchar(); printf (“c=%-4c”, c) ;
  i= getchar (); printf (“i=%-3d”, i) ;
  printf (“c1=%-4c”, getchar ( ) ) ;
}
```

‘a’ ‘b’ ‘c’ 回车符

97	98	99	10
----	----	----	----

在运行时，如果从键盘输入3个字符；abc（回车）。

运行结果：

c=a□□□i=98□c1=c

注意：上面getchar()函数被调用三次，分别得到字符‘a’、‘b’、‘c’，回车符还继续留在缓冲区里，如果再调用getchar()函数，得到的就是回车符。







## 3.6 顺序结构程序设计举例

例3.6.1 输入长方体的长、宽、高，求长方体的体积和表面积。

```
#include <stdio.h>
main()
{
float x,y,z,v,s;
printf("Input x,y,z:\n");
scanf("%f,%f,%f",&x,&y,&z);
v=x*y*z;
s=2*(x*y+x*z+y*z);
printf("x=%8.3f, y=%8.3f, z=%8.3f, v=%8.3f\n",x,y,z,v);
printf("s=%8.2f\n",s);
}
```

运行程序：

Input x,y,z

键盘输入 2.2,3.3,4.4 (回车)

屏幕显示为：

x=□□□2.200,y=□□□3.300,z=□□□4.400,v=□□□31.944

s=□□□62.92





## 3.6 顺序结构程序设计举例

**例3.6.2**编写并运行一个程序，打印出两个通过交互方式输入的整数的和、差、积、商及余数。

```
#include <stdio.h>
main( )
{ int m,n,a,b,c,d,e;
  printf("Input m,n:\n");
  scanf("%d%d",&m,&n);
  a=m+n; b=m-n; c=m*n; d=m/n; e=m%n;
  printf("m+n=%d\tm-n=%d\tm*n=%d\tm/n=%d\tm%%n=%d\n",
    a,b,c,d,e);
}
```

运行程序：

Input m,n:

键盘输入： 5 3（回车）

屏幕显示：

m+n=8 m-n=2 m\*n=15 m/n=1 m%n=2





## 3.6 顺序结构程序设计举例

**例3.6.3**求 $ax^2+bx+c=0$ 方程的根。

```
#include <stdio.h>
#include <math.h>
main()
{ float a,b,c,disc,x1,x2,p,q;
  printf("Input a,b,c:\n");
  scanf("%f%f%f",&a,&b,&c);
  disc=b*b-4*a*c;
  p=-b/(2*a);q=sqrt(disc)/(2*a);
  x1=p+q;x2=p-q;
  printf("x1=%5.2f\nx2=%5.2f\n",x1,x2);
}
```

运行程序：

Input a,b,c:

键盘输入： 1 1 -2 （回车）

屏幕显示：

x1=□ 1.00

x2=-2.00





# 附件：课程教师和助教（2011-2012第2学期）



## 主讲教师：林子雨

单位：厦门大学信息科学与技术学院计算机科学系  
办公地点：福建省厦门市思明区厦门大学海韵园  
E-mail: ziyulin@xmu.edu.cn  
个人主页：<http://www.cs.xmu.edu.cn/linziyu>

## 助教：林尚青

单位：厦门大学计算机科学系2010级硕士研究生  
E-mail: lsq1015@qq.com  
手机：15959206201

## 助教：赖明星

单位：厦门大学计算机科学系2011级硕士研究生  
E-mail: joy\_lmx@163.com  
手机：18050056577



# 附件：课程FTP（2011-2012第2学期）

- FTP地址：ftp://218.193.53.74
- 用户名：stu\_linziyu
- 密码：123456
- 目录：“下载教学内容”→“C语言”



# 附件：课程教材（2011-2012第2学期）

- 《C语言程序设计（第2版）》
- 清华大学出版社，黄保和，江弋 编著
- 版次：2011年10月第2版
- ISBN:978-7-302-26972-4
- 定价：35元

The background is a solid blue color with faint, light-blue silhouettes of people. At the top, there are two groups of people: one on the left holding hands in a circle, and one on the right standing in a line. On the right side, there is a large silhouette of a person talking on a mobile phone. In the bottom left corner, there are silhouettes of two people, one of whom appears to be holding a phone to their ear.

**Thank You!**

Department of Computer Science, Xiamen University, February 26, 2012