

# 高级语言程序设计

---

## Review

福州大学·软件学院·软件工程系·王灿辉  
电子邮件：[wangcanhui@fzu.edu.cn](mailto:wangcanhui@fzu.edu.cn)

# 考试概况

---

- Ø 考试方法：闭卷笔试
- Ø 考试时间：见教务网站
- Ø 考试地点：见教务网站
- Ø 考试时长：120分钟(2小时)
- Ø 总分：100分
- Ø 复习教案，完成作业

# 考题类型

---

考题类型	该题型题数	分值	该题型总分
单选题	20	1分/题	20
填空题	15	2分/空	30
程序填空	$\leq 5$	2分/空	20
编程题	$\leq 5$	5-15分	30

Ø 补考和期末考的题型、难度基本一致!

# 考题内容

---

- ❌ 授课内容：计算机语言和C概述、编程步骤和程序调试、顺序结构、选择结构、循环结构、枚举类型、**函数、数组和字符串**、结构与联合、位运算与位域、编译预处理、**指针和链表、文件管理**
- ❌ 考试要求：所有的章节都有题目覆盖。程序填空和编程题基本在函数、数组、字符串、文件、指针和链表这几章，其它几章的题目主要在选择和填空题中。

# 复习—结构化程序设计 (SP)

---

- Ø 模块化，分而治之；
- Ø 自顶向下，逐步求精；
- Ø 单入口，单出口；
- Ø 仅采用三种基本结构：顺序、选择/分支和循环；
- Ø 使用良好的编程风格；
- Ø 用栈和队列取代数组的使用；
- Ø 不用或少用GOTO语句。
- Ø .....

# 复习—算法和数据结构

---

- ❌ 每个程序都要依靠算法(查找：顺序法和二分法，排序：冒泡、选择、插入、快速等)和数据结构(数组、结构、指针、链表等)。
- ❌ 我们所需要的是知道有哪些合适的、可用的算法和数据结构，知道如何在各种可以互相替代的算法和数据结构之中做出选择。

# 复习—编程风格

---

- ❌ 代码应该是**清楚的和简单的**----具有直截了当的逻辑，自然的表达式，通行的语言使用方式，有意义的名字和有帮助的注解等。应该**避免耍小聪明**的花招，不使用非正规的结构。
- ❌ 很多人习惯于**遵守K&R的标准**，即参考资料2“C程序设计语言(第2版)”中的表达方式。

# 复习—编程风格

---

- Ø 从一开始就养成**良好的编程风格**：注解、缩进、命名等。同时注意一些惯例：底层检查并报告错误&高层处理错误；空间的申请和释放要在同一个层次进行处理；同样文件的打开和关闭也必须由同一层完成。
- Ø **标识符的作用域与其长度成正比！**使用范围越大名字就起得越长！



# 复习—编程风格

---

Ø 注解也是一种艺术，如：

```
zeroCount++;
```

```
//Increment zero entry counter
```

```
//这种注解纯粹是多此一举！
```

Ø 必须**保证注解和程序代码的一致**，否则比没有注解更坏！

Ø 标识符尤其是函数一定要和其含义相符！

Ø 给每个全局变量和函数加注解；不要注解差的代码，重写它！

# 复习—编程风格

---

- Ø 全局变量、函数、结构(类型)等必须使用说明性的长名字，最好加注解！
- Ø 局部变量尽量用短的名字，甚至单字母就足够了。否则(起长名)没有什么好处，甚至有害。
- Ø 采用自然的表达而不是程序(语句)越短越好；尽量不用否定的条件；适当加括号以避免二义性或错误。

# 复习—常量定义

---

Ø **不用神秘的数值！** 没有特殊原因，除0和1外的数值都必须定义成常量。

Ø **定义常量：**首选const，其次建议使用enum(只能定义整数值)，最后才是#define。

```
const int BUF_SIZE=80;
```

```
enum { BUF_SIZE=80};
```

```
#define BUF_SIZE 80
```

# 复习—排错 (纠错、调试)

---

- Ø 好的程序员知道他们在纠错上花费的时间至少与写程序一样多。
- Ø 你发现的任何错误都能教导你如何防止类似错误的再次发生，以及在发生这些问题时及早识别它。
- Ø 纠错是非常困难的，有可能花费很长的、无法预期的时间。
- Ø 早期预防胜过事后的治疗。

# 复习—排错 (纠错、调试)

---

- Ø 虽然语言可以避免发生错误，如：下标检查、限制指针使用、自动垃圾回收、字符串类型、带类型的IO、强类型检查等，但记住：没有任何语言能够防止你犯错误。
- Ø 有些语言特征有引起错误的倾向：全局变量、goto语句、无限制的指针及自动类型转换等等。
- Ø 建议：打开所有的编译检查，并留意所有的编译警告。

# 复习—排错 (纠错、调试)

---

- ❌ 实际程序设计的大部分时间确实花在了调试和排错上。
- ❌ 编译系统通常都带有一个复杂的排错系统。
- ❌ 大部分错误是非常简单的，很容易通过简单的技术找出来。
- ❌ **排错涉及到一种逆向推理**，就像侦破一个杀人迷案。
- ❌ 以单步方式遍历程序的方式，还不如努力思考，辅之于在关键位置上加输出语句和检查代码。

# 复习—排错手段

---

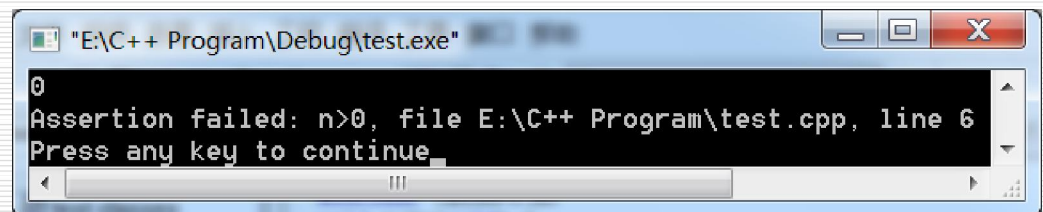
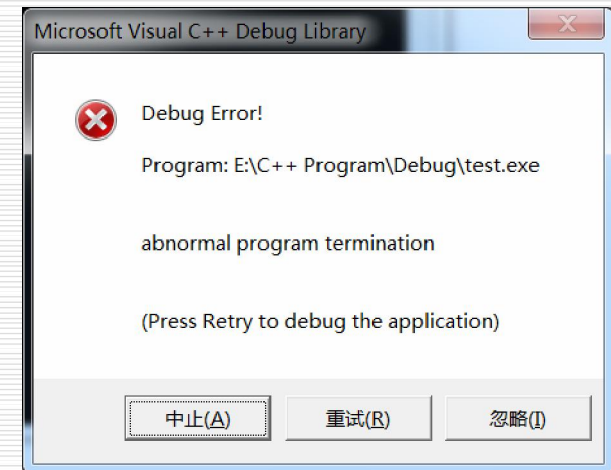
- ❌ 寻找熟悉的模式—少&、变量未初始化、错误的IO格式符、悬空指针等，检查最近的修改，不要两次犯同样的错误，现在就排除而不是等以后，键入前仔细想想—不要立即修改、打印程序关键部分、休息一下，把你的代码向他人解释。
- ❌ 重现错误，分而治之&局部化，研究错误的计数特征，写自检测代码，写日志文件等。
- ❌ 现在很多程序员已经无法享受全新开发的乐趣，他们的大部分时间用到维护上。

# 复习—保护性编程

- Ø 在程序里增加一些代码，专门处理所有“不可能”出现的情况。使用断言来进行程序前、后条件的测试。如：

```
#include <stdio.h>
#include <assert.h>
void main(void) {
    int n; scanf("%d",&n);
    assert(n>0);
    printf("%d\n",n);
}
```

- Ø 输入0，程序输出





# 复习—掌握和了解

---

- ❌ 课程虽然对C的绝大部分构造都进行介绍，但有些最好少用甚至不用。如：  
goto和continue语句、位域、条件编译、宏定义尤其是函数宏、全局变量、联合、类型转换尤其是自动类型转换、否定条件、复杂的表达式、下划线打头和C库函数名的标识符等。
- ❌ 这些内容会考，但分数少、难度低！

# 复习 计算机语言

---

- Ø 冯·诺依曼体系结构和IPOS
- Ø 计算机语言及分代
- Ø 计算机语言翻译过程—编译与解释
- Ø 计算机语言相关概念：源、目标、可执行程序及其扩展名；编辑、编译、链接和执行。

# 复习 : C 基本概念

---

- Ø C的产生和标准(作用), C的基本设计理念、特点和应用
- Ø C程序的基本结构
- Ø C的保留字(IF、printf等不是)和标识符
- Ø C语言的编程步骤
- Ø C语言程序的调试
- Ø 程序的算法和流程图

# 复习 基本数据类型

---

- Ø 实型、整型、字符型、字符串、各种数据类型的区别和类型转换等。
- Ø 数据类型占用空间，表示范围和溢出，有效位和误差，运算，常量格式和定义(符号常量)，变量定义和赋初值(初始化)，输入、输出格式
- Ø 数据类型是学习C的基础！

# 复习 常量

---

Ø `stdio.h`: `FILE`, `EOF`, `NULL`,  
`stdin`, `stdout`, `stderr`等

Ø `limit.h`: `INT_MIN`, `INT_MAX`,  
`UINT_MAX`, `LONG_MIN`等

Ø `HUGE_VAL`, `EXIT_SUCCESS`,  
`EXIT_FAILURE`, `RAND_MAX`

Ø .....

# 复习 其他数据类型

---

- Ø 枚举enum—定义、初始化和使用
- Ø 结构struct、联合union和位域
- Ø 数组(一至多维)和字符串(数组)
- Ø 指针: 概念, 声明, 赋值, 运算, 使用等
- Ø typedef和void类型
- Ø 结构+数组、结构+文件、结构+指针(链表)

# 复习 运算符和表达式

---

- 1) 算术运算符(+、-、\*、/、%、++、--)
- 2) 关系运算符(>、>=、<、<=、==、!=)
- 3) 逻辑运算符(与&&、或||、非!)
- 4) 位操作运算符(&、|、~、^、<<、>>)
- 5) 赋值运算符(=、复合算术和位运算赋值)
- 6) 条件运算符：三目运算符 (?:)
- 7) 逗号运算符，
- 8) 指针运算符：取内容(\*)和取地址(&)
- 9) 特殊运算符：**sizeof(提高程序可移植性)**，  
括号()，下标[]，成员(，.)等几种。

# 复习 运算符和表达式

---

- Ø 单目、二目、三目运算符
- Ø 优先级和结合性
- Ø 算术(数值)表达式、关系表达式和逻辑表达式、位运算
- Ø 赋值运算符和复合赋值符、赋值表达式和类型转换、逗号运算符和逗号表达式、求值顺序



# 复习 顺序语句

---

- Ø 赋值语句：`x = (a = 5) + (b = 8);`
- Ø 基本库函数的使用：库函数名、功能、参数和返回值、头文件等
- Ø 函数调用语句：`printf(...);`等
- Ø 输入：`scanf`函数和`getchar`函数
- Ø 输出：`printf`函数和`putchar`函数

# 复习：I/O格式

---

Ø 格式字符串的一般形式为：

**[标志][输出最小宽度][.精度][长度]类型**

Ø 标志(5个)：%-d,%0f...%**空格**f...,  
%+d,%#o,%#x,%#f...

Ø 宽度/精度：%5d,%5.1f...%\*d...

Ø 长度：%ld,%Lf,%hd,%hu ...

Ø 类型：%d,%i,%o,%x(X),%u,%f,  
%e(E),%g(G),%c,%s,%p,%%

# 复习 基本库函数

---

- Ø printf, scanf, putchar, getchar, sprintf, sscanf, remove等
- Ø strcpy, strcmp, strlen, puts, gets, strcat, strncmp, strncpy等
- Ø <ctype.h>: islower, toupper等
- Ø calloc, malloc, free, exit, system, abort, atexit, realloc等
- Ø .....

# 复习 数值类库函数

---

Ø abs, labs, fabs, div

Ø sqrt, pow, log, log10, exp

Ø floor, ceil

Ø sin, cos, tan, atan, .....

Ø rand, srand

Ø atoi, atol, atof

Ø .....

# 复习 文件操作类库函数

---

- Ø fopen, fclose, feof, fflush, ferror, clearerr, setbuf
- Ø fgetc/getc, fputc/putc, ungetc
- Ø fgets, fputs, fprintf, fscanf, fread, fwrite
- Ø fseek, ftell, rewind, 等.....

# 复习：I/O缓冲

---

- ❌ C标准只提供缓冲I/O方式，非缓冲I/O为非标准扩充的功能。
- ❌ 缓冲I/O又分完全缓冲(文件I/O使用，缓冲满时刷新)和行缓冲(键盘输入使用按回车键时刷新)。
- ❌ getchar, scanf等：行缓冲  
getch, getche等：非缓冲，非标准  
fprintf, fscanf等：完全缓冲

# 复习 变量

---

- ❌ 变量的命名(大小写、要有意义)、声明(先声明后使用)、初始化、赋值、运算、输入/输出等
- ❌ 变量的作用域(全局、局部、语句块和函数声明)和存储类型:  
static, auto, register, extern
- ❌ 变量的限定 : const, volatile

# 复习 选择结构

---

Ø if语句、if—else语句、if—else  
if—else语句、三目运算符、嵌  
套if语句、switch语句、复合语  
句{...}、空语句。



# 复习 循环结构

---

Ø 循环语句：while、do-while、for语句

Ø 多重循环

Ø 转移语句：break, continue  
goto, return语句

# 复习 函数

---

- Ø 函数的作用
- Ø 函数的声明、定义和调用
- Ø 形参和实参和参数的传递，函数的返回值，参数和返回值类型不匹配
- Ø 函数的嵌套和递归(调用)
- Ø 内部函数和外部函数

# 复习 数组和字符串

---

- Ø 数组(一至多维)类型的声明和引用
- Ø 数组的初始化
- Ø 数组的赋值、运算、输出和处理
- Ø 数组作为函数的参数传递
- Ø 字符数组&字符串
- Ø 常用字符串函数

# 复习：(编译)预处理

---

Ø #define(无参和有参宏)和#undef

Ø #include语句

Ø 条件编译语句:

#ifdef(#ifndef, #if)-#else-#endif

Ø C程序结构(多文件工程): 模块化的  
程序设计语言

# 复习 指针

---

- Ø 动态存储分配
- Ø 指针与数组，指针与字符串
- Ø 指针与函数
- Ø 命令行参数
- Ø 结构指针和链表(创建、遍历、插入、删除、合并、释放等)，文件和文件指针

# 复习 文件

---

- Ø 文件及其分类，文件指针
- Ø 与文件处理有关程序的程序框架：  
打开(和读写方式)  
读写和使用(IPO：读、处理和写)  
关闭。
- Ø 熟练掌握主要的一些库函数。如：  
fopen、fclose等

# 复习 :区别声明和使用

---

Ø `int a=5;`和`extern int x=5;` //全局

Ø `int a[5];`和`a[5]=5;`

Ø `int i, *p=&i;`和`*p=&i;`

Ø `int a,b;a=b=5;`和`int a=b=5;`

Ø `int a[2]={1,2};`和`a[2]={1,2};`

Ø `char *p="初始化字符串";`和  
`char *p;p="初始化字符串";`

Ø .....

# 区别函数声明、定义和调用

---

```
int max(int a,int b);           //函数声明
//或 : int max(int,int);       //函数声明
void main() {
    //.....
    z=max(x,y);                 //函数调用
    //.....
}
int max(int a, int b) {         //函数定义
    return (a>b?a:b);
}
```



# 复习 :区别函数声明、定义和调用

---

```
int func(int [], int);           //函数声明
```

```
//或:int func(int *,int);       //函数声明
```

```
void main() {
```

```
    //int a[5]=.....
```

```
    z=func(a,5);                 //函数调用
```

```
    //.....
```

```
}
```

```
int func(int a[], int n) { //函数定义
```

```
    //.....
```

```
}
```

# 复习—可移植性问题

---

- ❌ 可移植性和升级紧密相关，如：
- ❌ 一个程序依赖于早期版本的echo  
—如果改变规范就得改变名字！
- ❌ 有一个sum用于求文件的校验和(以便检查文件在传递中是否被改变)，但几台机器上使用的是不同的版本，这样等于没有任何意义。也就迫使大家都一起升级到最新的版本。

# 复习—可移植性问题

---

- Ø 数据类型的大小—用sizeof解决
  - Ø 求值顺序—除逻辑表达式外
  - Ø char的符号问题
  - Ø 算术或逻辑移位
  - Ø 字节顺序
  - Ø 结构或类成员的对齐
  - Ø 位域
- 这些在C标准中均没有明确的规定！

# 复习

---

- ❌ 计算的世界每时每刻都在变化，步伐越来越快。程序员必须不断应付新的语言、新的工具和新的系统，它们总有一些与老的东西不兼容的新特性。程序越来越大，界面越来越复杂，而任务的时限越来越短。
- ❌ 让我们高举程序主义、软件工程思想的伟大旗帜，紧密团结在以MS为核心的软件公司周围，沿着比尔·盖茨的生财之道，不分白天黑夜地编程，把建设有中国特色的软件产业的伟大事业全面推向21世纪。

# 复习 基本算法

---

∅ 考题主要涉及：

- 1、基本概念及语法题
- 2、阅读程序选择或给出输出结果
- 3、程序功能、程序的互为选择
- 4、指出/改正程序中的错误，不完整程序填空等。

∅ 必须掌握的一些常用的基本算法：求和/平均、计数、求阶乘、日期处理、判素数、求最大、最小值、排序、查找、递归、链表、输出特殊图形、用迭代法计算近似值等。

# 复习 典型题目

---

- 1、判断合法性？
- 2、代数表达式和C表达式的相互转换；
- 3、(算术、关系、逻辑)表达式的求值；
- 4、判断各种叙述的正确性？
- 5、程序功能、程序的等价性选择；
- 6、阅读程序选择或给出输出结果；
- 7、判断、指出、改正程序中的错误；
- 8、不完整程序填空等。

# 复习 注意事项

---

- ❌ 由于题目量相对比较大，面较广，所以复习必须全面。
- ❌ 题目相对比较简单，太偏或太难的题目很少。
- ❌ 重点章节必须掌握（如：数组、函数等），有些章节只需掌握其主要部分。

# 样题：选择题

---

Ø 以下符号中，合法的用户标识符是：  
(B)。

A . #pp      B . \_page\_no\_  
C . int      D . 3x

Ø 以下选项中，与 $m=n++$ 完全等价的表达式是：(D)。

A.  $m=++n$       B.  $m+=n+1$   
C.  $n=n+1, m=n$       D.  $m=n, n=n+1$



# 样题：选择题

---

Ø 在C90标准下，下列语句中，能正确定义一维数组的是：(B)。

- A . `int N=10;`      B . `#define N 10`  
          `int a[N];`                      `int a[N];`  
C . `int a[0-10];`    D . `int a[];`

# 样题：选择题

---

Ø 下列程序片段错误的是：(A)。

A . int a=b=c=0;

B . int a=0,b=0,c=0;

C . int a,b,c; a=b=c=0;

D . int a,b,c; a=0,b=0,c=0;

# 样题：选择题

---

❌ 与循环语句 `for (int i=0,s=0;i<10;i++,s+=i);` 等价的语句或语句组是：(C)。

- A . `int i,s;`  
    `s=i=0;`  
    `do {`  
        `s+=i++;`  
    `}`  
    `while (i<10);`
- B . `int i=s=0;`  
    `for (;i<10;i++) s+=i;`
- C . `int i=0,s=0;`  
    `while (i<10) { s+=++i;}`
- D . `int s=0;`  
    `for (int i=0;i<10;s+=i++);`

# 样题：填空题

---

下述程序的输出结果是： 7。

```
#include <stdio.h>
int f() {
    static int c=5;
    return (++c);
}
void main() {
    int k;
    for(int i=0;i<2;i++) k=f();
    printf("%d\n",k);
}
```

# 样题：填空题

---

下述程序的输出结果是：[11][20][21][11]。

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

```
int x=10;
```

```
void func() {
```

```
    int x=20;
```

```
    printf("[%d]",x);
```

```
    x++;
```

```
    printf("[%d]",x);
```

```
}
```

```
void main() {
```

```
    x++;
```

```
    printf("[%d]",x);
```

```
    func();
```

```
    printf("[%d]\n",x);
```

```
}
```

# 样题：填空题

---

Ø 移动文件内部位置指针的函数  
是：\_\_\_\_\_。

fseek

17

Ø 执行以下程序后输出：\_\_\_\_\_。

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

```
#define MUL(x) x*x
```

```
void main() {
```

```
    int a=5;
```

```
    printf("%d\n",MUL(a+2));
```

```
}
```

# 样题：填空题

---

Ø 以下程序的功能  
为：\_\_\_\_\_。

字符串拷贝

```
void f(const char* from,char *to)
{
    while (*from)
        *to++ = *from++;
    *to = '\0';
}
```

# 样题：程序填空题

---

Ø 判n为素数，是返回真否则返回假

```
int isPrime(int n) {  
    int k=(int)sqrt(n);  
    for (int i=2;i<=k;i++)  
        if ( n%i==0) return false;  
    return true;  
}
```



# 样题：编程和综合题

---

- Ø 编写函数convert将字母替换为\*，保留数字，删除其他字符。例如：字符串"This 99 !\*#!",处理后结果为："\*\*\*\*99"。
- Ø 判一个字符串是否为回文串？
- Ø 输出所有3位数的对称素数。例如：101、727都是素数并且是对称的。
- Ø 用递归法实现求n!，反序输出字符串，求最大公约数和菲波拉契数列等程序。

# 样题：编程和综合题

---

- ❌ 编写矩阵运算的相关函数：矩阵的加、减、乘、转置等。
- ❌ 分别用数组、结构数组、链表实现简单的学生成绩管理系统（成绩输入、成绩分析与统计、排序、查找、成绩输出等功能）。
- ❌ 将文件内容读出、显示并进行各种统计和分析。

# 复习到此结束！

---



福州大学·软件学院·软件工程系·王灿辉  
电子邮件：[wangcanhui@fzu.edu.cn](mailto:wangcanhui@fzu.edu.cn)