



# 高级语言程序设计 C++

张海威  
南开大学信息技术科学学院

# 期末复习

<http://it.nankai.edu.cn>

1

题型

2

复习知识点

3

考试题目示例

# 考试题型

<http://it.nankai.edu.cn>

- ❖ 单项选择
- ❖ 程序改错
- ❖ 读程序写结果
- ❖ 程序填空
- ❖ 程序设计



# 期末复习

<http://it.nankai.edu.cn>

1

题型

2

复习知识点

3

考试题目示例

## ❖ 程序的SP框架结构

- 预处理指令
- 全局变量
- 函数原型与函数定义
- 主函数
- 注释

## ❖ 各部分的有序组合，构成SP框架结构

# 复习知识点

<http://it.nankai.edu.cn>

- ❖ 数制转换
- ❖ C++语言的词汇
- ❖ 数据类型、运算符与表达式
- ❖ C++语言的语句
- ❖ 一维数组、多维数组、字符数组
- ❖ 结构类型
- ❖ 函数
- ❖ 运算符重载
- ❖ 全局变量、局部变量、静态变量



# C++语言的词汇

<http://it.nankai.edu.cn>

❖ 关键字

❖ 标识符

- 合法的标识符

❖ 字面常量

- 四类，特别注意整型常量前后缀、浮点常量后缀

❖ 运算符

❖ 分割符



# C++语言的词汇

<http://it.nankai.edu.cn>

❖ 以下有4个用户自定义标识符，其中合法的一个是( **A** )。

A. C\_7

B. do

C. 7d

D. for



# C++语言的词汇

<http://it.nankai.edu.cn>

❖ C++程序中，字面常量012对应的二进制数与下列哪个二进制数等值（ A ）

A. 1010

B. 10010

C. 1100

D. 1101

## ❖ 基本类型

## ❖ 基本类型的派生类型

## ❖ 用户定义类型

- 枚举类型：定义方式、与整数的对应关系
- 结构类型：类型定义方式与变量定义方式

## ❖ 导出类型

- 数组类型
- 引用类型：只作为函数参数考查

# 运算符与表达式

<http://it.nankai.edu.cn>

- ❖ 运算符含义
- ❖ 运算分量
- ❖ 运算表达式
  - 逻辑运算表达式的短路问题
- ❖ 多种运算的表达式
  - 运算符的优先级与结合性



# 运算符与表达式

<http://it.nankai.edu.cn>

## ❖ 运算中的隐式数据类型转换（也称为数据类型的兼容）

- 整型、字符型、布尔型、枚举型之间的兼容性
- 整型和浮点型的兼容性
- 单精度浮点型与双精度浮点型的兼容性



# 运算符与表达式

<http://it.nankai.edu.cn>

❖ 在VC上执行下列语句后，a的值是( A )。

```
int a =10;
```

```
a-=++a*a;
```

A. -110      B. -120      C. -99      D. -109

# 运算符与表达式

<http://it.nankai.edu.cn>

❖ 设有代码 “`int a=3;`”，执行语句 “`a+=a--*a`” 后，a的值是（ B ）。

A. 6                  B. 11                  C. 10                  D. 12

# 运算符与表达式

<http://it.nankai.edu.cn>

❖ 设有变量说明 “`int x=5, y=6;`”，执行表达式语句 “`!(++x)&&(y*=y);`” 后，`x`和`y`的值分别为（ **D** ）

A. `x=5, y=6`

B. `x=6, y=36`

C. `x=5, y=36`

D. `x=6, y=6`



- ❖ 说明语句
- ❖ 表达式语句
- ❖ 控制语句
  - 分支语句
  - 循环语句
  - 转向语句
- ❖ 复合语句（语句块）
- ❖ 空语句



❖ 下面的C++关键字中，哪个不能用于变量的说明语句（ **B** ）

A. extern

B. inline

C. static

D. int

❖ 若 $x=100$ , 则与`while(x)`语句等价的语句是  
( D )。

A. `while(x==0)`

B. `while(x==100)`

C. `while(x!=100)`

D. `while(x!=0)`

❖ `int i=0, j=2, k=4;` 下列选项中错误的if语句是 ( B )。

A. `if (i==j) i=k++;`

B. `if (k=>j) i=k++;`

C. `if (j-k) k-=j++;`

D. `if (i) --i;`

❖ 若i, j已定义为整型变量, 则以下程序段中循环体的执行次数是 ( B )。

```
for (i=4; i ; i--)  
    for (j=0; j<6 ;j++)  
        { 循环体语句; }
```

A) 20      B) 24      C) 35      D) 30

# C++语句

<http://it.nankai.edu.cn>

```
#include <iostream.h>
void main()
{
    int a, b;
    for(a=1, b=1; a<=20; a++)
    {
        if (b%3==1) {
            b+=3;
            continue;
        }
        if(b>10) break;
    }
    cout<<"value a="<<a<<"    value
b="<<b<<endl;
}
```



# C++语句

<http://it.nankai.edu.cn>

```
#include <iostream.h>
void main()
{
    int i,j;
    char a;
    for(i=5; i>=1; i--)
    {
        for(j=1; j<=5-i; j++)
            cout<<' ';
        a='*';
        for(j=1; j<=2*i-1; j++)
            cout<<a;
        cout<<endl;
    }
}
```



# C++语句

<http://it.nankai.edu.cn>

```
#include <iostream.h>
void main()
{
    int i,j;
    char a;
    for(i=1; i<=5; i++)
    {
        cout<<"          ";
        for(j=5; j<=1; j++)
            cout<<'  ';
        a='*';
        for(j=1; j<=2*i-1; j++)
            cout<<a;
        cout<<endl;
    }
}
```



# C++语句

<http://it.nankai.edu.cn>

```
#include<iostream.h>
void main() {
    char ch;
    for(int i=0;i<5;i++){
        for(int j=1;j<5-i;j++)
            cout<<" ";
        for(int k=65;k<71-j;k++){
            ch = k;
            cout<<ch;
        }
        for (k=k-2;k>64;k--) {
            ch = k;
            cout<<ch;
        }
        for(int m=0;m<5-i;m++)
            cout<<" ";
        cout<<endl;
    }
}
```





# C++语句

<http://it.nankai.edu.cn>

```
#include<iostream.h>
void main()
{
    int n=5;
    for(int i=1;i<=n;i++)
    {
        int result = 1;
        for(int j=1;j<=i;j++) ;
        {
            result *=j;
        }
        cout<<"result = "<<result<<endl;
    }
}
```



❖ 下面这个程序用于读入5个整数，当程序读入的数据为正整数时，则显示该数，否则，程序结束运行

# C++语句

<http://it.nankai.edu.cn>

```
#include <iostream.h>
void main()
{
    int i, n;
    for (i=1; i<=5; i++)
    {
        cout<<"Please enter the integer number:\n";

        _____
        if (n <= 0) _____
        _____
    }
    cout<<"Program is over!\n";
}
```



❖ 输入一个字母，如果是小写字母，则把它变成大写字母输出；如果是大写字母，则把它变成小写字母输出；其它字符不变。请完善程序。

```
#include<iostream.h>
void main()
{
    char ch;
    cin>>ch;
    if ( _____ (1) _____ )
        if(ch>='a' && ch<='z')
            _____ (2) _____;
        else
            _____ (3) _____;
    cout<<ch<<endl;
}
```

❖ 已知能被4整除而不能被100整除的或者能被400整除的年份是闰年，则判断某一年是否是闰年

# C++语句

<http://it.nankai.edu.cn>

```
#include <iostream.h>
void main()
{
    int year, leap;
    cout<<"please input the year"<<endl;
    cin>>year;
    if ( _____ )
        leap=1;
    else _____
        if( _____ )
            cout<<"是闰年"<<endl;
        else cout<<"不是闰年"<<endl;
}
```



❖ 利用公式  $\frac{\pi}{2} = \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \dots$  前100项之积，编程计算并输出  $\pi$  的值

```
#include <iostream.h>
void main()
{
    float term, result = 1.0;
    int n;
    for (_____)
    {
        term = _____;
        result = result * term;
    }
    cout << _____
}
```

❖ 求出50以内所有能被3整除的整数的平方和并输出

```
#include <iostream.h>
```

```
void main() {
```

```
    for(int i=2; i<50 ;                     )
```

```
        s +=                      ;
```

```
    cout<<"s="<<s<<endl;
```

```
}
```



## ❖ 一维数组和二维数组

- 说明方式
- 初始化
- 访问

## ❖ 字符数组

- 说明方式
- 初始化方式
  - 按照字符串进行初始化
  - 将字符数组整体进行输入输出
- 访问字符或子字符串

# 数组

<http://it.nankai.edu.cn>

❖ 已知: `int a[3][5]`; 则对数组元素的非法引用是 ( C )。

A. `a[0][2*1]`

B. `a[0][0]`

C. `a[3][0]`

D. `a[0][4]`

# 数组

❖ 可以用数组名作为字符串变量整体输出的合法的数组说明语句是（ A ）。

A) `char a[]="string";`

B) `int a[5] = {0, 1, 2, 3, 4, 5};`

C) `char a= "string";`

D) `char a[ ]=  
{ '1' , ' 2' , ' 3' , ' 4' , ' 5' };`

❖ 在下面的数组定义中，有语法错误的是  
( C )。

A. `int a[] = {1, 2, 3, 4};`

B. `int a[10] = {0};`

C. `int a[];`

D. `int a[5];`

# 数组

<http://it.nankai.edu.cn>

❖ 若二维数组a有m列，则在a[i][j]之前的元素个数为：       B      。

- A)  $j*m+i$
- B)  $i*m+j$
- C)  $i*m+j-1$
- D)  $i*m+j+1$



# 数组

❖ 下面能正确定义一个int型二维数组a的语句是 ( B ) 。

- A. `int a[][]={2, 3, 0, 0, 4, 0, 5, 0, 0, 0, 0};`
- B. `int a[][4]= {2, 3, 0, 0, 4, 0, 5, 0, 0, 0, 0};`
- C. `int a[3][4]= {{2, 3}, {4, , 5}};`
- D. `int  
a[][]={{2, 3, 0, 0}, {4, 0, 5, 0}, {0, 0, 0, 0}};`

# 数组

- ❖ 下面对于二维数组的说明及初始化，不正确的是（ A ）
- ❖ `char a[][10]={"C++","programming"};`
- ❖ `int a[2][5] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};`
- ❖ `float a[2][5] = {1.0, 5.5, 9.9};`
- ❖ `double a[][5] = {1, 2.0, 3, 4.5, 5.5};`

# 数组

<http://it.nankai.edu.cn>

```
#include <iostream.h>
void main()
{
    int i,j=0;
    char a[50];
    cin >>a;
    for(i=0; a[i]!='\0'; i++)
    {
        if
        (! (a[i]>='a' && a[i]<='z' || a[i]>='A' && a[i]<='
        Z')) j++;
    }
    cout<<j<<endl;
}
```





# 数组

<http://it.nankai.edu.cn>

```
#include <iostream.h>
void main() {
    int i,j;
    char a[10]="bacdfgeih";
    for(i=0;i<9;i++) {
        for(j=i;j<9;j++) {
            char t;
            if (a[i]>a[j]) {
                t=a[i];
                a[i]=a[j];
                a[j]=t; }
        }
    }
    a[9]='\0';
    cout<<a<<endl;
}
```



# 数组

<http://it.nankai.edu.cn>

```
#include <iostream.h>
int power(int m);
void main() {
    int a[3][3], i, j;
    for (i=0; i<3; i++) {
        for (j=0; j<3; j++) {
            a[i][j]=(i+2)*(j+1)+1;
            cout<<"    "<<a[i][j];
        }
        cout<<endl;
    }
}
```

# 数组

<http://it.nankai.edu.cn>

```
cout<<"----执行结果----"<<endl;
int m=a[0][0], s=0;
for (i=0; i<3; i++)
    for (j=0; j<3; j++) {
        if ( a[i][j]<m ) m=a[i][j];
        if ( i!=j )
            s+=power(a[i][j]);
        cout<<"m="<<m<<endl;
        cout<<"s="<<s<<endl;
    }
int power(int m) {
    return m*=m;
}
```



# 字符数组

<http://it.nankai.edu.cn>

❖ 下面函数实现strcmp的功能，将两个字符串s和t进行比较，如果相等返回0，否则返回两个字符串中第一个不相同字符的ASCII码值之差。

```
int MyStrcmp(char s[], char t[])
{
    int i;
    for (i=0; _____ ; i++)
    {
        if (_____) return 0 ;
    }
    _____
}
```



# 字符数组

<http://it.nankai.edu.cn>

❖ 任意输入一个字符串，在星期名表中查找该字符串，若找到，则打印“已找到”，否则打印“没找到”

```
#include <iostream.h>
#include <string.h>
void main() {
    int i;

    char x[10];
    char weekDay[7][10]=
        {"Sunday", "Monday", "Tuesday",
        "Wednesday", "Thursday", "Friday", "Saturday"};
    cout<<"请输入一个字符串:";
```



# 字符数组

<http://it.nankai.edu.cn>

```
for (i=0; i < 7 && !findFlag; i++)
{
    if (strcmp(x, weekDay[i]) == 0)
        findFlag = 1;
}
if _____
cout<<"已找到\n";
else cout<<"没找到!\n";
}
```



# 字符数组

<http://it.nankai.edu.cn>

❖ 下列函数实现将字符串str的内容逆序存放到ptr中（假设两者长度相等）

```
#include <string.h>
void Inverse(char str[], char ptr[]) {
    int i = 0, j;

    while (_____) {
        ptr[i] = str[j];
        i++;
        j--;
    }

    _____
}
```



# 字符数组

<http://it.nankai.edu.cn>

❖ 函数Count () 计算字符串str1在字符串str中出现的次数。请完善该函数

```
int Count(char str[], char str1[])
{
    int i,j,k,num=0;
    for (i=0; _____ (1) _____; i++)
        for (_____ (2) _____, k=0; str1[k]==str[j]; k++, j++)
            if (str1[_____ (3) _____] == '\0')
            {
                num++;
                i+=k;
                break;
            }
    return num;
}
```





## ❖ 字符串匹配的条件

- 从第一个相同字符开始，依次比较
- 当短串结束时，如果短串中全部字符以相同顺序包含于长串中，则匹配成功一次
- 设计思路：
  - 找到长串与短串第一个相同字符
    - 二重循环：外层长串不结束，内层  $s[j]==t[k]$
  - 开始循环，每次向后移动一个位置进行比较，如果相同则继续向后移动（直到短串结束），如果不同，则结束本次匹配，在长串中寻找下一个与短串第一个字符相同的字符

# 结构类型

<http://it.nankai.edu.cn>

- ❖ 结构类型的含义
- ❖ 结构类型的定义
- ❖ 结构类型变量的说明与初始化
- ❖ 结构类型变量成员的访问方式
- ❖ 结构类型的举例：作业



# 结构类型

<http://it.nankai.edu.cn>

- ❖ 电影《2012》中提到的历法来自于玛雅文明。玛雅文明是最早发明、使用历法的文明之一。玛雅人日常生活使用的是哈布历（Haab'）。哈布历规定一年365天，分为19个月，前18个月每月20天，第19个月5天。玛雅文明还使用一种用于宗教祭祀的卓尔金历（Tzolkin，就是这个历法预言2012年的灾难）。卓尔金历规定一年260天，分为13个月，每个月20天。哈布历和卓尔金历的公元0年1月1日是同一天开始的（考古学家认为是公元前3114年8月31日），请编写一个程序实现哈布历和卓尔金历的相互转换，要求：
  - 如果输入一组（至少5个）哈布历日期，将其转换为卓尔金历日期（日期以日 月 年的顺序表示），并按照日期由小到大的顺序输出
  - 如果输入一组（至少5个）卓尔金历日期，将其转换为哈布历日期（日期以日 月 年的顺序表示），并按照日期由小到大的顺序输出



# 结构类型

<http://it.nankai.edu.cn>

例如：

输入历法（H表示哈布历，T表示卓尔金历）： H

8	11	2002
13	3	1996
7	16	2563
16	1	835
1	9	245

输出历法卓尔金历（T），排序后结果为：

6	8	344
11	4	1172
13	4	2802
18	4	2811
2	4	3599

# 结构类型

<http://it.nankai.edu.cn>

```
#include <iostream.h>
```

```
const int N = 5;
```

```
struct date{  
    int year;  
    int month;  
    int day;  
}calendar[N];
```

```
void sort_calendar(date[],int) ;  
void swap(date&,date&);
```



# 结构类型

<http://it.nankai.edu.cn>

```
void main() {  
    char ch;  
    int day ,mon, year;  
    long sum;  
    cout << "please input catelog of calendar: ";  
    cin>>ch;
```



# 结构类型

<http://it.nankai.edu.cn>

```
if( (ch=='H') || (ch=='h'))           { //哈布历法
    for(int i=0;i<N;i++){
        cin>>calendar[i].day>>calendar[i].month>>calendar[i].year;
        day = calendar[i].day;
        mon = calendar[i].month;
        year = calendar[i].year;
        sum= day+mon*20+year*365;
        year=sum/260;
        sum=sum % 260;
        mon=sum/20;
        day=sum%20;
        calendar[i].day = day;
        calendar[i].month = mon;
        calendar[i].year = year;
    }
    cout<<"Output calendar of Tzolkin" <<endl;
}
```



# 结构类型

<http://it.nankai.edu.cn>

```
else{
    if ((ch=='T') || (ch=='t'))    {
        for(int i=0;i<N;i++)    {
            cin>>calendar[i].day>>calendar[i].month>>calendar[i].year;
            day = calendar[i].day;
            mon = calendar[i].month;
            year = calendar[i].year;
            sum= day+mon*20+year*260;
            year=sum/365;
            sum=sum % 365;
            mon=sum/20;
            day=sum%20;
            calendar[i].day = day;
            calendar[i].month = mon;
            calendar[i].year = year;
        }
        cout<<"Output calendar of Haab'"<<endl;
    }
    else
        cout <<"input error"<<endl;}
```





# 结构类型

<http://it.nankai.edu.cn>

```
sort_calendar(calendar,N) ;//日期排序
for(int i=0;i<N;i++) //结果输出
    cout<<calendar[i].day<<" "<<calendar[i].month<<"
"<<calendar[i].year<<endl;
}
```



# 结构类型

<http://it.nankai.edu.cn>

```
void sort_calendar(date d[],int n){
    for(int i=0;i<n-1;i++)
        for(int j=i;j>0;j--)
            if(d[j].year<d[j-1].year)
                swap(d[j],d[j-1]);
            else
                if(d[j].year==d[j-1].year)
                    if(d[j].month<d[j-1].month)
                        swap(d[j],d[j-1]);
                    else
                        if(d[j].month==d[j-1].month)
                            if(d[j].day<d[j-1].day)
                                swap(d[j],d[j-1]);
} //交换函数定义略
```



## ❖ 函数的说明方式

- 函数原型，一般写在主函数之前
- 函数定义，一般写在主函数之后，若无原型，一般写在主函数之前

## ❖ 函数的调用

- 嵌套
- 递归

## ❖ 函数的返回值

## ❖ 函数的参数

- 引用类型参数的意义

# 函数

<http://it.nankai.edu.cn>

```
#include <iostream.h>
void f(int x, int &y, const int &z)
{
    x+=z;
    y+=z;
    cout<<"x="<<x<<" , y="<<y<<" , z="<<z<<endl;
}
void main()
{
    int a=22,b=33,c=44;
    f(a,b,c);

    cout<<"a="<<a<<" , b="<<a<<" , c="<<c<<endl;
    int d=b-c;
    f(a+b,d,2*c-10);
}
```



# 函数

<http://it.nankai.edu.cn>

❖ 函数Square () 测试输入的参数是否是完全平方数，如是返回1 否则返回0（例如对1, 4, 9, 16, 25返回1）。请完善该函数。

```
int isSquare(int n)
{
    _____ (1) _____;
    while ( _____ (2) _____ )
        ++i;
    if ( _____ (3) _____ ) return 1;
    else return 0;
}
```

# 函数

<http://it.nankai.edu.cn>

- ❖ 1742年6月7日，哥德巴赫写信给欧拉，提出了以下猜想：任何一个大于等于6的偶数都可以表示成为两个奇素数之和；任何一个大于等于9的奇数都可以表示成为三个奇素数之和——这就是著名的哥德巴赫猜想。
  - 请编写程序验证在10000以内哥德巴赫猜想是否成立。如果成立输出Yes否则输出No



## ❖ 判断素数

```
int isprime(int n) {  
    //判定素数函数，如果素数返回1，否则返回0  
    for(int i=2;i<n/2;i++)  
        if(n%i==0)  
            return 0;  
    return 1;  
}
```

# 函数

<http://it.nankai.edu.cn>

```
int even(int n) {  
    //判定一个大于等于6的偶数是否符合哥德巴赫猜想  
    for ( int i=3;i<n;i+=2)    {  
        if(isprime(i)&&isprime(n-i))    {  
            return 1;  
        }  
    }  
    return 0;  
}
```





# 函数

<http://it.nankai.edu.cn>

```
int odd(int n) {  
    //判定一个大于等于9的奇数是否符合哥德巴赫猜想  
    for ( int i=3;i<n;i+=2)    {  
        if(isprime(i) && even(n-i)) {  
            return 1;  
        }  
    }  
    return 0;  
}
```

# 函数

<http://it.nankai.edu.cn>

```
void main() {    //主程序 在6到10000, 9到10000做
    循环分别判定偶数和奇数
    bool g1=true,g2=true;
    for(int i=6;i<=10000;i+=2)
        g1=g1 && even(i);
    for(i=9;i<=10000;i+=2)
        g2 = g2 && odd(i);
    if(g1&&g2)
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;
}
```



## ❖ 字符数组与字符串

- 统计字符、“短”字符串在某个“长”字符串中出现的次数、每次出现时字符或“短”字符串的下标
- 按照某些规则，将一个或几个字符串进行变换，得到新的字符串
  - 注意，在表示新字符串的字符数组的适当位置设置 `'\0'`，作为串尾符
- 字符串的输出，包括正序、逆序、指定字符的输出等等

## ❖ 运算符重载函数

- 原型
- 定义
- 使用方式：参见教材P145bool类型运算符重载例子

## ❖ 变量各种存储类型的含义

### ■ 全局变量与局部变量

- 生存期与作用域

### ■ 静态变量

- 初始化一次
- 作用域为块级
- 生存期为程序级



The End