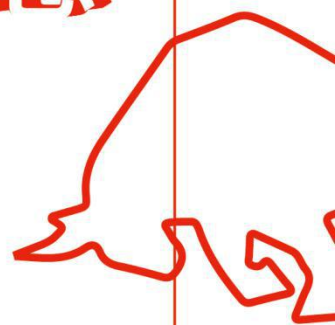
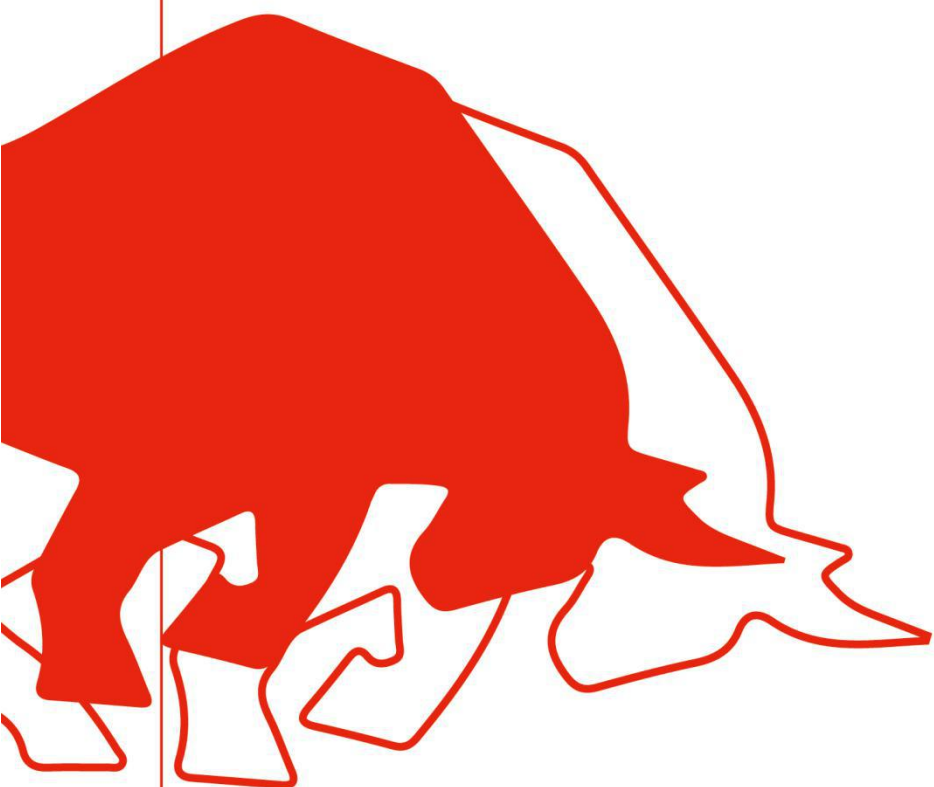




# 主观题汇总

C++程序设计



## 使用说明：

1. 此主观题汇总文档是按题型整理的，而题型来自于对历年真题的总结。参考 1804 考期，本科目主观题总分为 60 分，各题型分值分布为：①程序分析结果题：2\*5 分=10 分；②程序填空题：5\*4 分=20 分；③程序设计题：1\*10 分=10 分；④程序改错题：5\*4 分=20 分。
2. 所有知识点分高中低三个频次，以该知识点被考察次数和最新考试大纲为依据进行排序。
3. 每道题前数字表示曾经被考到的年份和考期，比如 1804，表示该题目在 2018 年 4 月份被考到。没有数字表示的为模拟题。

# 程序分析结果题汇总

## 高频知识点：

### 一、静态成员

1. ( 1110 ) 请给出下面程序的输出结果

```
#include<iostream>
using namespace std ;
void fun()
{
    static int n=25 ;
    n-- ;
    cout<<"n="<<n<<endl;
}
void main()
{
    for(int i=0 ; i<=2 ; i++)
        fun() ;
}
```

答案：

**n=24**

**n=23**

**n=22**

2. ( 1404 ) 程序分析题

```
#include<iostream . h>
void func( ) ;
void main( )
{
    for(int i=0 ; i<6 ; i++)
    {
        func( ) ;
    }
}
```

```

void func( )
{
int x=0 ;
x ++ ;
static int y=0 ;
y ++ ;
cout<<"x="<<x<<" ; y="<<y<<endl ;
}

```

答案：

**x=1;y=1**

**x=1;y=2**

**x=1;y=3**

**x=1;y=4**

**x=1;y=5**

**x=1;y=6**

## 二、派生类的构造函数和析构函数

1. ( 1510 ) 程序分析题。

```

#include <iostream.h>
class Base
{ private:
int Y;
public:
Base(int y=0) { Y=y;cout <<" Base(" <<y <<" )\n";}
~ Base() { cout <<" ~ Base()\n";}
void print() { cout <<Y <<" ";}
};
class Derived:public Base
{ private:
int Z;
public:
Derived ( int y, int z):Base(y)
{ Z=z;
cout <<" Derived(" <<y <<" , " <<z <<" )\n";
}
~ Derived() { cout <<" ~ Derived()\n";}
}

```

```

void print()
{ Base::print();
cout << Z << endl;
}
};
void main()
{ Derived d(10,20);
d.print();
}

```

答案：

**Base(10)**

**Derived(10,20)**

**10 20**

**~Derived()**

**~Base()**

2. ( 1604 ) 程序分析题。

```

#include <iostream>
using namespace std;
class CBase
{
public:
    void fun();
};
void CBase::fun()
{ cout << "调用基类类的函数 fun() \n";
}
class CDerived:public CBase
{
public:
    void fun();
};
void CDerived::fun()
{
    cout << "调用派生类的函数 fun() \n";
}
void main()
{
    CDerived d1;
    CBase *pb = &d1;
    CBase &pd = d1;
    d1.fun();
    pb -> fun();
    pd.fun();
}

```

答案：

运行结果：

调用派生类的函数 fun( )

调用基类类的函数 fun( )

调用基类类的函数 fun( )

### 三、使用类的对象

1. ( 1310 ) 给出下面程序的输出结果

```
#include <iostream>
using namespace std;
class base
{
int x ;
public :
void setx( int a){x=a;}
int getx ( ){return x ; }
} ;
void main ( )
{
int*p;
base a ;
a.setx (15) ;
p= new int (a. getx( ) ) ;
cout < < * p;
}
```

答案：15

### 四、类模板的成分及语法

1. ( 1210 ) 写出此程序的执行结果：

```
#include <iostream.h>
template<class T>
T func(T x , T y){
if(sizeof(T)==8)
```

```

return x+y ;
else
return x*y ;
}
void main(){
cout <<func(8 , 5)<<" : "<<func(5.0 , 10.0)<<endl ;
int d ; cin>>d :
}

```

答案：40:15

## 五、单一继承的一般形式

1. ( 1810 ) 阅读程序后，填写程序的正确运行结果。

```

#include <iostream. h >
using namespace std;
class Sample{
protected:
    int x;
public:
    Sample() { x = 0; }
    Sample( int val ) { x = val; }
    void operator++ ( ) { x++ ; }
};
class Derived:public Sample{
    int y;
public:
    Derived() :Sample( ) { y = 0; }
    Derived( int val1 ,int val2 ):Sample( val1 ) { y = val2; }
    void operator -- ( ) { x-- ;y-- ; }
}

```

```

void disp() {
cout << " x = " << x << " , y = " << y << endl; }
};
void main() {
    Derived d(3,5);
    D. disp();
    d++;
    D. disp();
    d--;
    d--;
    D. disp();
}

```

答案：

**x=3,y=5**

**x=4,y=5**

**x=2,y=3**

## 中频知识点：

### 一、认识 C++ 的函数和对象

#### 1. ( 0910 ) 程序分析题

```

#include <iostream>
using namespace std ;
int main() {
    int i,j ;
    for(i=1 ; i<=4 ; i+=1)
    for(j=2 ; j<=4 ; j+=j){
        if(i+j<5)
            cout<<" + " ;
        else
            cout<<" * " ;
    }
    return 0 ;
}

```



}

答案：+ \* + \* \* \* \*

## 二、使用输出和输入对象

1. ( 1510 ) 给出下面程序输出结果。

```
#include "iostream. h"
void main( )
{
    int x,y,z;
    x = 3;
    y = x ++ ;
    z = ++ x;
    cout << "x = " << x << " ; y = " << y << " ; z = " << z << endl;
}
```

答案：

x=5;y=3;z=5

## 三、使用 string 类的典型成员函数实例

1. ( 1410 )

请写出 myText1.txt 文本文件中的内容

```
#include <iostream>
#include <string>
using namespace std;
#include <fstream>
void main()
{
    ofstream myFile1;
    myFile1.open("myText1.txt");
    cout << "Enter the data in Chinese format( e. g. ,2008,May 25):" << endl;
    string Date("2008,January 1");
    string Year = Date.substr(0,4);
    int k = Date.find(",");
    int i = Date.find(" ");
    string Month = Date.substr(k + 1, i - k - 1);
    string Day = Date.substr(i + 1, 2);
    string NewDate = Day + " " + Month + " " + Year;
    myFile1 << "original date:" << Date << endl;
    myFile1 << "Converted date:" << NewDate << endl;
    myFile1.close();
}
```

答案：

original date:2008,January 1

Converted date:1 January 2008

#### 四、函数的参数及其传递方式

##### 1. ( 1704 ) 运行结果题

```

#include <iostream>
using namespace std;
void sort(int L[ ], int n)
{
    int j, k, flag, temp;
    flag = n - 1;
    while(flag > 0)
    { k = flag - 1; flag = 0;
      for(j = 0; j <= k; j++)
          if(L[j] > L[j + 1])
              { temp = L[j]; L[j] = L[j + 1];
                L[j + 1] = temp; flag = j; }
    }
}

void main()
{ int array[4] = {7, 2, 3, 4};
  sort(array, 4);
  cout << "The sorted numbers: ";
  for(int i = 0; i < 4; i++)
      cout << array[i];
}

```

答案：

**The sorted numbers : 2347**

## 五、引入函数模板

1. ( 1101 ) 给出下面程序的输出结果。

```

#include<iostream>
using namespace std ;
template<class T>
T max(T m1 , T m2)
{return(m1>m2)?m1 : m2 ; }
void main() {

```

```
cout<<max(1 , 7)<< " \ t"<<max(2.0 , 5.0)<<endl ;
cout<<max( 'y' , 'b')<< " \ t"<<max("A" , "a")<<endl ;
}
```

答案：

**7 5**

**y a**

## 低频知识点：

### 一、定义成员函数

1. ( 1410 )

```
Simple( ) { x = y = 0; }
Simple( int i,int j) { x = i;y = j; }
void copy( Simple &s) ;
void setxy( int i,int j) { x = i;y = j; }
void print( ) { cout << "x = " << x << " ,y = " << y << endl; }
};
void Simple::copy( Simple &s)
{
x = s. x;y = s. y;
}
void func( Simple s1 ,Simple &s2)
{
s1. setxy( 30,40) ;
s2. setxy( 70,80) ;
}
void main( )
{
Simple obj1( 1,2) ,obj2;
obj2. copy( obj1) ;
func( obj1 ,obj2) ;
obj1. print( ) ;
obj2. print( ) ;
}
```

答案：

**x=1 , y=2**

**x=70 , y=80**

### 二、自动调用构造函数

1. ( 1001 ) 请给出下面程序的输出结果

```
#include < iostream. h >

class A
{
public:
A();
void Show() ;
~A();
private:
static int c;
};

int A : : c =0;
A::A()
{
cout < < "constructor." < < endl;
c+ =10;
}
void A: :Show()
{
cout< < "c=" < <c< <endl;
}
A : : ~A()
{
cout < < "destructor. " < < endl;
}
void main()
{
A a , b ;
a. Show() ;
b. Show() ;
}
```

答案：

**constructor.**

constructor.

c=20

c=20

destructor.

destructor.

## 程序填空题汇总

高频知识点：

### 一、静态成员

1. ( 1310 ) 在下面程序的底藏线处填上适当的字句，使该程序执行结果为 40，X 为静态成员。

```
#include <iostream. h >
class Test
{ public:
    _____;
    Test ( int i = 0)
    { x = i + x; }
    int Getnum( )
    { return Test::x + 10; }
    } ;
    _____;
void main( )
{ Test test;
  cout << test. Getnum( ) << endl;
}
```

答案:

第 1 空：static int x ;

第 2 空：int Test::x=30;

## 二、类模板的成分及语法

1. ( 1410 )

在下面下划线处填上正确的语句，使程序输出结果为：

201

402

3

4

5

```
#include <iostream>
using namespace std;
template < class T >
T f( T &a, T &b, int n) {
    for( int i = 0; i < n; i ++ ) {
        a[i] = a[i] + b[i];
    }
    return a;
}
void main( ) {
    int a[5] = { 1,2,3,4,5 };
    int b[5] = { 100,200,300,400,500 }, *p;
    for( int i = 0; i < 2; i ++ ) {
        p = ①;
    }
    for( ; p < a + 5; p ++ ) {
        cout << ② << endl;
    }
}
```

答案:

第 1 空：f(a,b,2)

第 2 空：\*p

## 三、使用输出和输入对象

1. ( 1610 ) 完成程序空白处，可参考以下运行结果：

4 !=24

n=-2 1 不能计算 n!

程序执行结束

```
#include <iostream.h>
int f(int n)
{
    if (n <=0) throw n;
    int s = 1;
    for( _____ )
        s *= i;          //计算 1 * 2 * 3 ... * n 的值
    return s;
}
void main( )
{
    try
    {
        cout<< _____ << endl;
        cout<< " -2! ="<< f(-2)<< endl;
    }
    catch( int n ) { cout<< "n ="<< n<< "不能计算 n!"<< endl; }
    catch( ... ) { cout<< "出现异常!"<< endl; }
    cout<< "程序执行结束"<< endl;
}
```

答案:

第1空: `int i=1;i<=n;i++`

第2空: `"4!="<<f(4)`

#### 四、声明类

1. ( 1504 )



```

Sample:: Sample( int a)
{
x = a;
y++ ;
}
void Sample::print( )
{
cout << "x = " << x << " ,y = " << y << endl;
}
_____ ;
void main( )
{
Sample s1(5);
Sample s2(10);
s1.print();
s2.print();
}

```

答案:

第 1 空 : **public**

第 2 空 : **int Sample::y=25**

## 五、定义成员函数

1. ( 1404 ) 完成下面类中的成员函数的定义。

```

class point
{
private:
int m, n;
public:
point(int, int);
point(point &);
};
point:: point(int a, int b)
{
m=a;
_____=b;
}
point:: point(_____)
{
m=t. m;
n=t. n;
}

```

答案:

第 1 空 : **n**

第 2 空 : **point &t**

## 六、使用类的对象

1. ( 1410 )

完成下面类的定义，使其可以正确的初始化成员变量 m 和 n:

```
class point {
private: int m,n;
public:
point( int,int );
point( point &t );
};
point::point( int a, int b ) {
    m = a;
    ① = b;
}
point::point( ② ) {
    m = t. m;
    n = t. n;
}
```

答案:

第 1 空 : **n ;**

第 2 空 : **point &t**

## 七、单一继承的一般形式

1. ( 1704 ) 程序填空题

```
#include <iostream>
using namespace std;
class F
{ public:
    _____
};
double F::operator()( double x,double y) const
{
    return (x+5)*y;
}
_____
{ F f;
    cout << f(1.5,2.2) << endl;
}
```

答案:

第 1 空 : **double operator( )(double x,double y) const;**

第 2 空 : **void main( )**

## 八、使用文件流

1. ( 1504 ) 下面是一个输入半径，输出其面积和周长的 C++ 程序，在下划线处填上正确的语句。

```
#include "stdafx. h"
#include <iostream>
#include "math. h"

_____

void main( )
{ double rad;
  cout << " rad = " ;
  cin >> rad;
  double l = 2. 0 * pi * rad;
  double s = pi * rad * rad;
  cout << " \n The long is:" << l << endl;
  cout << "The area is:" << s << endl; }
```

答案:

第 1 空 : **using namespace std;**

第 2 空 : **#define pi 3.14159**

## 九、构造函数的定义和使用方法

1. ( 1810 ) 请按空格顺序填写答案。

```

完成程序,使其结果为
x = 6, y = 10
x = 11, y = 10
#include <iostream. h >
using namespace std;
class Sample {
private:
    int x;
    _____;
public:
    Sample( int a );
    void print();
};
Sample::Sample( _____ )
    { x = a; y = x++ ; }
void Sample::print()
    { cout << "x = " << x << " , y = " << y << endl; }
int Sample::y = 25;
void main()
{
    Sample s1(5);
    Sample s2(10);
    s1. print();
    s2. print();
}

```

答案：

第一空：**static int y**

第二空：**int a**

## 十、认识 C++ 的函数和对象

1. ( 1810 ) 请按空格顺序填写答案。

完成程序,使其输出结果为 79

```
#include <iostream.h>
using namespace std;
int main() {
    int a[6] = {23,15,64,33,40,58};

    _____
    s1 = s2 = a[0];
    for( int *p = a + 1; p < a + 6; p++ ) {
        if( s1 > *p ) s1 = *p;
        if _____;
    }
    cout << s1 + s2 << endl;
    return 0;
}
```

答案：

第一空：int s1,s2;

第二空：(s2<\*p)s2=\*p

## 中频知识点：

### 一、const 修饰符和预处理程序

1. ( 1610 ) 完成程序空白处

```

#include <iostream>
using namespace std;

_____ PI = 3.1415;
double length (float r)//圆的周长
{
    return 2 * PI * r;
}

_____//矩形的周长
{
    return 2 * (x + y);
}
void main()
{
    float a = 2, b = 3, r = 4;
    cout<< "圆周长"<< length(r)<< endl;
    cout<< "矩形周长"<< length(a, b)<< endl;
}

```

答案:

第1空: **const double**

第2空: **double length (float x, float y)**

## 二、使用 complex 对象

1.(1101)在下划线处填上缺少的部分。

```

#include<iostream>
#include<fstream>
using namespace std ;
class complex
{
public :
int real ;
int imag ;
complex(int r=0 , int i=0)
{
real=r ;
imag=i ;
}

```

```
};
complex operator+(_____, complex& b)
{
    int r=a.real+b.real ;
    int i=a.imag+b.imag ;
    return_____ ;
}
void main()
{
    complex x(1 , 2) , y(3 , 4) , z ;
    z=x+y ;
    cout<<z.real<<"+"<<z.imag<<"i"<<endl ;
}
```

答案:

第 1 空：**complex& a**

第 2 空：**complex(r,i)**

### 三、对象作为函数参数

1.(1604)补充程序。

```
#include <iostream>
using namespace std;
int a[ ] = {2,4,6,7,10};
int &index( int i)
{   return a[i]; }
void main( )   {
    int i;
    _____ //调用 index 函数,将数组 a 中的元素 7 替换为 8。
    for ( _____ ) //输出数组 a
        cout << a[i] << " ";
}
```

答案:

第 1 空：**index(3)=8;**

第 2 空：**i=0;i<=4;i++**

### 四、引用作为函数参数

1.(1310)在答题纸上填上缺少的部分。源程序如下：

```
#include <iostream>
using namespace std;
class base
{
int a,b;
public :
base( int x,int y){a=x;b=y;}
void show ( _____ )
{
cout<<p.a<<" , "<<p.b<<endl;
}
}_____
void main( )
{
base b(78,87) ;
b.show(b) ;
}
```

答案:

第 1 空：**const base &p**

第 2 空：；

## 低频知识点：

### 一、派生类的构造函数和析构函数

1. ( 1410 )

填空使程序的输出结果如下：

Parent::Parent( )called.

Child::Child( )called.

Child::~~Child( )called.

Parent::~~Parent( )called.



```

#include <iostream>
using namespace std;
class Parent
{
public:
    Parent() { cout << "Parent::Parent() called. \n"; }
    virtual ~Parent() { cout << "Parent::~~Parent() called. \n"; }
};
class Child:public Parent
{
public:
    Child( int i ) {
        cout << "Child::Child() called. \n";
        buf = new char[i];
    }
    virtual ~Child() {
        delete [ ] buf;
        cout << "Child::~~Child() called. \n";
    }
private:
    char * buf;
};
void disp( ① * a )
{
    ②;
}
void main( )
{
    Parent * a = new Child(20);
    disp( a );
    int d;
    cin >> d;
}

```

答案:

第 1 空 : **Parent**

第 2 空 : **delete a**

## 二、重载对象的赋值运算符

1. ( 1504 ) 在下列程序的空格处填上适当的字句，使输出为：0，2，10。

```

#include "stdafx.h"
#include <iostream>
#include "math.h"
using namespace std;
class Magic
{ double x;
public:
Magic( double d = 0.00 ) : x( fabs( d ) )
{ }
Magic operator + ( _____ )
{
return Magic( sqrt( x * x + c. x * c. x ) );
}
_____ << ( ostream & stream, Magic & c )
{ stream << c. x;
return stream;
}
};
void main( )
{ Magic ma;
cout << ma << " , " << Magic(2) << " , " << ma + Magic( -6 ) +
Magic( -8 ) << endl;
}

```

答案:

第 1 空 : Magic &c

第 2 空 : friend ostream&operator

## 程序设计题汇总

高频知识点：

### 一、声明类

1. ( 1810 )

设计一个圆类 circle 和一个桌子类 table。circle 类包含私有数据成员 radius 和求圆面积的成员函数 getarea(); table 类包含私有数据成员 height 和返回高度的成员函数 getheight()。roundtable 类继承所有上述类的数据成员和成员函数, 添加了私有数据成员 color 和相应的成员函数。其中, main 函数已给出。请完成程序的其他部分。

```
void main()
{
    roundtable rt(0.8, 1.2, "黑色");
    cout << "圆桌属性数据" << endl;
    cout << "高度:" << rt.getheight() << "米" << endl;
    cout << "面积:" << rt.getarea() << "平方米" << endl;
    cout << "颜色" << rt.getcolor() << endl;
}
```

答案:

```
#include <iostream.h>
#include <string.h>
using namespace std;
class circle{
    double radius;
public:
    circle(double r) { radius = r; }
    double getarea() { return radius * radius * 3.14; }
};
class table{
    double height;
public:
    table(double h) { height = h; }
    double getheight() { return height; }
};
class roundtable: public table, public circle{
    char * color;
public:
    roundtable(double h, double r, char c[]) { circle(r); table(h);
        color = new char[strlen(c) + 1];
        strcpy(color, c);
    }
    char * getcolor() { return color; }
};
void main()
{
    roundtable rt(0.8, 1.2, "黑色");
    cout << "圆桌属性数据" << endl;
    cout << "高度:" << rt.getheight() << "米" << endl;
    cout << "面积:" << rt.getarea() << "平方米" << endl;
    cout << "颜色:" << rt.getcolor() << endl;
}
```

2. (1210) 设计一个 Bank 类, 实现银行某账号的资金往来账目管理, 包括建账号、存入、取出等。解: Bank 类包括私有数据成员 top(当前账指针), date(日期), money(金额), rest(余

额)和 sum(累计余额)。有三个成员函数 bankin()(处理存入账), bankout()(处理取出账)和 disp()(输出明细账)请完成函数的实现。

本题程序如下：

```
#include<stdio.h>
#include<string.h>
#define Max 100
class Bank
{
int top ;
char date[Max][10] ; //日期
int money[Max] ; //金额
int rest[Max] ; //余额
static int sum ; //累计余额
public :
Bank(){top=0 ; }
void bankin(char d[ ] , int m){...}
void bankout(char d[ ] , int m){...}
void disp(){...} ;
} ;
```

答案：

1 )

```
strcpy(date [top],d);
money[top]=m;
sum=sum+m;
rest[top]=sum;
top++;
```

2)

```
strcpy(date[top],d);
money[top]=-m;
sum=sum-m;
rest[top]=sum;
top++;
printf(“日期存入 取出余额\n” );
```

3)

```
int i;
for(i=0;i<top;i++)
{
printf("%8s",date[i]);
if(money[i]<0)
printf("%6d",-money[i]);
else
printf("%6d",money[i]);
printf("%6d\n",rest[i]);
}
```

## 二、认识 C++ 的函数和对象

1. ( 1310 )

在字符串类 string 中实现一个判断函数，该函数功能是统计某一字符串类对象（仅有单词和空格组成）有多少个单词，同时保存所有单词在字符串中的起始地址（设该字符串不超过 100 个单词）

```
#include <iostream.h>
#include <string>
class str{
string s;
int n,a[100],j,l;
public:str(string&a){s=a;n=0;j=0;l=0;}
... test(...);
int* geta(){return a;}
};
```

请写出 test 函数的过程（如果需要形式参数，请给出形参类型和数量，以及返回值类型）

答案：

```
inline void str::test()
{
int k=0;
for(int i=0;i<s.size();i++)
{
if(s[i]!='&&i!=0)
```

```

        {
            a[n]=k;
            n++;
            k=i+1;
        }
    }
    if(s[s.size()-1]!='')
    {
        a[n]=k;
        n++;
    }
}

```

### 三、函数的参数及其传递方式

#### 1. ( 1404 )

在三角形类 tri 实现两个函数，功能是输入三个顶点坐标判断是否构成等边三角形

```

#include<iostream . h>
#include<math . h>
class point { point
private : float x , y ;
public : f(float a , float b){x=a ; y=b ; }
f( ){x=0 ; y=0 ; }
void set(float a , float b){x=a ; y=b ; }
float getx( ){return x ; }
float gety( ){return y ; }
};
class tri{
point x , y , z ;
float s1 , s2 , s3 ;
public....settri(...); //用于输入三个顶点坐标
....test(...); //用于判断是否构成等边三角形
};

```

请写出两个函数的过程(如果需要形式参数，请给出形参类型和数量，以及返回值类型)

答案：

```
void tri::test(){
s1=sqrt((x.getx()-y.getx())*(x.getx()-y.getx())+(x.gety()-y.gety())*(x.gety()-y.gety()));
s2=sqrt((x.getx()-z.getx())*(x.getx()-z.getx())+(x.gety()-z.gety())*(x.gety()-z.gety()));
s3=sqrt((y.getx()-z.getx())*(y.getx()-z.getx())+(y.gety()-z.gety())*(y.gety()-z.gety()));
if(s!=s3&& s2==s1)
cout<<"三个顶点坐标能构成等边三角形";
else cout<<"三个顶点坐标不能构成等边三角形" ; }
void tri::settri(float x1,float y1,float x2,float y2,float x3,float y3)
{x.set(x1,y1);y.set(x2,y2);z.set(x3,y3);}
```

## 中频知识点：

### 一、使用 complex 对象

#### 1. ( 1504 )

声明复数类,Complex,该类中有两个私有变量 real, image 分别表示一个复数的实部和虚部。为 Complex 类添加适当的构造函数。并使用友元函数 add 实现复数加法。

```
#include <iostream>
using namespace std;
class Complex
{
private:
double real, image;
public:
void setRI( double a, double b)
{
real = a;
image = b;
}
```

```

double getReal()
{
    return real;
}
double getImage()
{
    return image;
}
void print() {
    if (image > 0)
        cout << "复数:" << real << " + " << image << "i" << endl;
    if (image < 0)
        cout << "复数:" << real << " - " << image << "i" << endl;
}
friend Complex add(Complex, Complex); //声明友元函数
};

void print() {
    if (image > 0)
        cout << "复数:" << real << " + " << image << "i" << endl;
    if (image < 0)
        cout << "复数:" << real << " - " << image << "i" << endl;
}
friend Complex add(Complex, Complex); //声明友元函数
};

void main()
{
    Complex c1(19, 0.864), c2, c3;
    c2.setRI(90, 125.012);
    c3 = add(c1, c2);
    cout << "复数一:"; c1.print();
    cout << "复数二:"; c2.print();
    cout << "相加后:"; c3.print();
}

```

答案：

在类内添加：

**Complex() { }**

**Complex(double a, double b)**

```

{
    real = a;
    image = b;
}

```

类外添加：



```

Complex add(Complex c1,Complex c2)//定义友元函数
{
Complex c3;
c3.real=c1.real+c2.real;//访问 Complex 类中的私有成员
c3.image=c1.image+c2.image;
return c3;
}

```

## 二、单一继承的一般形式

### 1. ( 1510 )

写一个程序，定义一个抽象类 Shape，由它派生 3 个类：Square(正方形)、Trapezoid(梯形)和 Triangle(三角形)。用虚函数分别计算几种图形面积、并求它们的和。要求用基类指针数组，使它每一个元素指向一个派生类对象。

```

#include <iostream.h>
class Shape
{ public:
virtual double area() const = 0;
};

```

答案：

```

class Square:public Shape
{
public:
    Square(double s) : side(s){ }
    double area() const{return side *side;}
private:
    double side;
};

class Trapezoid:public Shape
{
public:
    Trapezoid(double i,double j,double k):a(i),b(j),h(k)

```

```

    {}
    double area( )const{ return((a+b)*h/2);}
private:
    double a,b,h;
};
class Triangle:public Shape
{
public:
    Triangle(double i,double j):w(i),h(j)
    {}
    double area( )const{return(w*h/2);}
private:
    double w,h;
};
void main( )
{
    Shape *p[5];
    Square se(5);
    Trapezoid td(2,5,4);
    Triangle te(5,8);
    p[0]=&se;
    p[1]=&td;
    p[2]=&te;
    double da=0;
    for(int i=0;i<3;i++)
    {da+=p[i]->area();}
    cout<<"总面积是 : "<<da<<endl;
}

```

### 三、派生类的构造函数和析构函数

#### 1. ( 1604 )

定义一个图形类 ( figure ) , 其中有保护类型的成员数据 : 高度 ( height ) 和宽度 ( width ) , 一个公有的构造函数。由该图形类建立两个派生类:矩形类和等腰三角形类。在每个派生类中都包含一个函数 area( ), 分别用来计算矩形和等腰三角形的面积。其中 , 主要函数已有 ,

请完成其他部分的代码。

```
int main( )
|
triangle tri(2,3);
rectangle rec(2,3);
cout << "The area of triangle is:" << tri.area() << endl;
cout << "The area of rectangle is:" << rec.area() << endl;
return 0;
|
```

答案：

```
#include<iostream>
using namespace std;
class figure
{
protected:
    double height,width;
public:
    figure(double=0,double=0);
}
figure::figure(double h,double w)
{
    height=h;
    width=w;
}
class triangle:public figure
{
public:
    double area( );
    triangle(double=0,double=0);
};
triangle::triangle(double h,double w):figure(h,w)
{
    height=h;width=w;
```

```

}
double triangle::area( )
{
    return 0.5*height*width;
}
class rectangle:public figure
{
public:
    double area();
    rectangle(double=0,double=0);
};
rectangle::rectangle(double h,double w):figure(h,w)
{
    height=h;
    width=w;
}
double rectangle::area()
{
    return height*width;
}

```

#### 四、类模板的成分及语法

##### 1. ( 1010 )

定义队列类模板 queue(先进先出) , 队列的大小由使用者确定。要求该类模板对外提供如下二种基本操作：(1)insert(入队列)(2)front(出队列) , 用数组来实现

```

#include < iostream >
using namespace std;
template < class T, int size >
class queue {
    T x [ size ] ;
    int current, front, rear;
public :
    queue ( ) { current = 0 ; front = 0 ; rear = 0 ;}

```

```
...insert(...);
...front(...);
};
```

请写出两个函数的过程(如果需要形式参数,请给出形参类型和数量,以及返回值类型)

答案:

```
void queue<T,size>::insert(T t){
if(current==size)
{cout<<"queue overflow!"<<endl;exit(0);}
}
T queue<T,size>::front(){
if(current==0){
cout<<"deleting from an empty queue!"<<endl;exit(0);}
else{T a=x[front];current--; (front++)%size;return a;}
}
```

## 低频知识点:

### 一、类模板的派生与继承

#### 1. (1410)

定义堆栈类模板 Stack(先进后出),栈的大小由使用者确定。要求该类模板对外提供如下二种基本操作:

(1)push 入栈(2)pop 出栈,用数组来实现

```
#include <iostream>
using namespace std;
template < class T, int size >
class Stack{
T x[ size ];
int current;
public:
Stack() { current = 0; }
... push(...);
... pop(...);
};
```

请写出两个函数的过程(如果需要形式参数,请给出形参类型和数量,以及返回值类型)

答案:

```
void Stack<T,size>::push ( T t ) {
if(current==size){
cout<<"The Stack is full!"<<endl;exit(0);}
```

```

else{x[current]=1;current++;}
}
T Stack<T,size>::pop ( ) {
if(current==0){
cout<<"There is no object in the Stack!"<<endl;exit(0);}
else{T a=x[current-1];current--;return a;}
}

```

## 程序改错题汇总

### 高频知识点：

#### 一、函数的参数及其传递方式

1. ( 1810 , 0901 )

下面的类定义中有一处错误，请用下横线标出错误所在行及错误原因。

```

#include<iostream.h>
class f{
private : int x , y ;
public : f1(int a , int b){x=a ; y=b ; }
void print(){cout<<x<<y<<endl ; }
} ;
main(){
f a;
a.f1(1.5 , 1.8) ;
a.print() ;
}

```

答案：

**a.f1(1.5,1.8); f1( )调用时实参与形参类型不一致。**

2. ( 1310 ) 下面的每题中有一处错误，请在“答题纸”上将错误所在行写出并给出错误原因。

```

#include < iostream. h >
class f{

```

```

private: int x,y;
public: f1( int a,int b) { x = a;y = b; }
void print() { cout << x << --<< y << endl; }
};
main() {
f a;
float x = 1.0,y = 2. 0;
a. f1( x,y) ;
a. print() ;
}

```

答案：

**float x=1.0 , y=2.0 ; f1( )调用时实参与形参类型不一致。**

## 二、认识 C++的函数和对象

1. ( 1810 , 1310 ) 下面的每题中有一处错误，请在“答题纸”上将错误所在行写出并给出修改意见。

```

#include < iostream. h >
class test{
private: int x;y;
public: voicl f( int a,int b) { x = a;y = b; }
int max( ) { return(x > y) ? x:y; }
};
main() {
test a;
a. f( 1 ,3) ;
cout << a. max( ) << endl;
}

```

答案：

**int x;y; x 定义时使用了 ; 号 , y 没有定义。**

**应改 int x,y ;**

2. ( 1110 ) 下面的每题中有一处错误，请用下横线标出错误所在行及错误原因。

```

#include < iostream. h >

```

```

void main(){
float * p;
p = new float[ 5 ];
for( i =0;i <5 ;i + + )
{ cin > > * (p + i) ;cout << * (p + i) << endl; }
delete [ ] p;
}

```

答案：

**for(i=0;i<5;i++)变量 i 必须先定义后使用**

### 三、构造函数的定义和使用方法

1. ( 1704 )

程序改错题

以下是对类 Sample 的定义

```

#include <iostream>
using namespace std;
class Sample
{
public:
    Sample( int val );
    ~ Sample( ) ;
private:
    float a =2.5;
    Sample( ) ;
};

```

答案：

**Sample(int val);改为 Sample(float val);**

### 四、左值和右值

1、( 1404 ) 下面的每题中有一处错误，请在答题纸上写出错误所在行并给出错误原因

```

#include<iostream . h>
main ( ){
int x=5 , y=6 ;
const int *p= &x ;
* p=y ;
cout<<*p<<endl ;

```



}

答案：

**\*p=y ; 指向常量的指针 p 不能进行左值操作。**

## 五、数据成员的赋值

1. ( 1101 ) 下面的类定义中有一处错误,请用下横线标出错误所在行并指出错误原因。

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

```
class f{
```

```
private : int x=0 , y=0 ;
```

```
public : void f1(int a , int b){x=a ; y=b ; }
```

```
void get(){cout<<x<<' , '<<y<<endl ; }
```

```
};
```

```
main(){
```

```
f a ;
```

```
a.f1(1 , 3) ;
```

```
a.get() ;
```

```
}
```

答案：

**private:int x=0,y=0;数据成员声明时不允许直接赋值**

## 六、静态成员

1. (1510)下面的程序段中有一处错误，请在答题卡上抄写错误所在行并指出错误原因。

```

#include <iostream>
using namespace std;
class A {
    static int x;
    int y;
public:
    A(int a, int b) {
        x = a;
        y = b;
    }
    int get() {
        return x + y;
    }
};
x = 5;
int main()
{
    A a(1, 2);
    cout << a.get() << endl;
    return 0;
}

```

答案：

**x=5；错。x 是静态数据成员，该赋值形式错误。**

**修改：int A::x=5；**

## 七、类模板的成分及语法

1. (1504)下面的程序段中各有一处错误，请将错误所在行抄写在答题卡上，并指出错误原因。

```
#include <iostream. h >
class f{
private: T x,y;
public: f1(T a,T b){x = a;y = b;}
float max( ) {return(x > y)? x:y;}
};
main() {
f a;
a. f1(1.5,3.8);
cout < < a. max( ) < < endl;
}
```

答案：

T x , y ;

模板未声明。

## 八、析构函数和运算符 delete

1. ( 1810 ) 以下程序中有一处错误，请抄写有错误的语句并改正。

申请一个长度为 10 的 int 型空间，之后释放该空间。

```
#include <iostream. h >
using namespace std;
int main( ) {
    int * p = new int[ 10 ] ;
    delete p;
    return 0;
}
```

答案：

delete p;错误，改为 delete []p;

## 九、使用输出和输入对象

1. ( 1810 ) 以下程序中有一处错误，请抄写有错误的语句并改正。

```
4. #include <iostream.h>
using namespace std;
class A {
    int x,y;
public:
    void set( int a,int b)
        { x = a;y = b; }
    int getx()
        { return x; }
    int gety()
        { return y; }
    int sum()
        { return x + y; }
};
int main() {
    A a;
    a.set(3,4);
    cout << a.x << a.y << a.sum() << endl;
    return 0;
}
```

答案：

**cout<<a.x<<a.y<<a.sum()<<endl;**      **错      误      ,      改      为**  
**cout<<a.getx()<<a.gety()<<a.sum()<<endl;**

## 中频知识点：

### 一、声明类

1. (0810)下面的类定义中有一处错误，请用下横线标出错误所在行及错误原因

```
#include<iostream.h>
template<class T>
class f{
private : T x , y ;
```

```

public : void f1(T a , T b){x=a ; y=b ; }
T max( ){return(x>y)?x:y;}
};
main(){
f a ;
a.f1(1.5,3.8) ; '
cout<<a.x<<a.y<<a.max( )<<endl ;
}

```

答案：

**cout<<a.x<<a.y<<a.max( )<<endl;x,y 是私有类型，不能在主函数直接使用。**

## 二、使用类的对象

1. (1001)下面的每题中有一处错误，请用下横线标出错误所在行及错误原因。

```

#include < iostream. h >
class Class
{
private :
int val ;
public :
void func( int a)
{
val = a;
}
};
void main()
{
Class * cls ;
cls - > func(10) ;
}

```

答案：

**cls->func(10);指针 cls 在调用前没有初始化**

## 三、函数模板的参数

1. (1210)下面的类定义中有一处错误，请写出错误所在行并给出修改意见。

```
#include<iostream.h>
template<class T>
T minx(T x , T y){
if(x>y)
x=y ;
return x ;
}
void main()
{
int a ;
double d ;
cout<<minx(a , d)<<endl ;
}
```

答案：

**minx 函数的参数类型必须一致 int a 或 double b**  
**将 a 修改为 double 类型，或将 d 修改为 int 类型。**

#### 四、使用类的权限

1. (1101)下面的类定义中有一处错误,请用下横线标出错误所在行并指出错误原因。

```
#include<iostream.h>
class point{private : float x ;
public : void f(float a){x=a ; }
void f(){x=0 ; }
friend float max(point& a , point& b) ;
} ;
float max(point& a , point& b)
{return(a.x>b.x)?a.x : b.x ; }
main(){
point a , b ;
a.f(2.2) ; b.f(3.3) ;
cout<<a.max(a , b) ;
}
```

答案：

`cout<<a.max(a,b);max()`不是类的成员函数，不能用 `a.max()` 方式调用。

## 五、常量成员

### 1. (1604)改错题

```
#include <iostream>
using namespace std;
void main()
{
    int x1(3),x2(8);
    int * const p = &x1;
    cout << *p << endl;
    p = &x2;
    cout << *p << endl;
}
```

答案：

`int * const p=&x1;`错误

修改 `int *p=&x1 ;`

## 六、派生类支配基类的同名函数

### 1. (1404)下面的每题中有一处错误，请在答题纸上写出错误所在行并给出修改意见

```
#include<iostream . h>

class test1{
private : int x ;
public : test1( ){x=2 ; }
void set(int a){x=a ; }
void get(){cout<<x<<endl ; }
};

class test2{
private : int x ;
public : test2( ){x=3 ; }
void set(int a){x=a ; }
void get( ){cout<<x<<endl ; }
};

class test : public test1 , public test2 . {
private : int x ;
```

```

public : void set(int a){x=a ; }
void gettest( ){cout<<x<<endl ; }
};
main( ) {
test a ;
a . get( ) ;
}

```

答案：

**a.get( );**

**get( )存在二义性，使用时需要指出使用哪个基类的函数。**

## 七、纯虚函数与抽象类

1. (1210)下面的类定义中有一处错误，请写出错误所在行并给出修改意见。

```

#include <iostream.h>
class Base{
public : virtual void fun()=0 ;
} ;
class Test : public Base{
public : virtual void fun(){cout<<"Test.fun="<<endl ; }
} ;
void main(){
Base a ;
Test *p ; p= &a ;
}

```

答案：

**Base a ; 抽象类不能被实例化。**

## 低频知识点：

### 一、引用

1. (1110)下面的每题中有一处错误，请用下横线标出错误所在行及错误原因

```

#include < iostream, h >
class f{

```



```

private: float x, y;
public: void f1 ( float a, float b) { x = a; y = b; }
f() {x=0;y=0;}
void move(float a,float b){x =x +a;y =y-b;}
void get ( ) { cout << x <<"<< y << endl; }
};
void main(){
f *p,a;
a. f1(1.6,3.8) ;
p=&a;
p. get( ) ;
}

```

答案：

**p.get( );p 作为指针引用时使用 “->”**

## 二、指向常量的指针

1. (1010)下题中有一处错误，请用下横线标出错误所在行并给出修改意见

```

#include < iostream, h >
void main() {
int x =5;
const int * const p = &x;
* p =88;
cout << * p << endl;
}

```

答案：

**\*p=88；指向常量的常量指针不能进行左值或赋值操作。**

## 三、使用 string 对象

1. (1010)下题中有一处错误，请用下横线标出错误所在行并给出修改意见

```

#include < iostream, h >
void main( ) {
string str1 (" we are here" );
cout << str1<< endl;
}

```

```
}
```

答案：

**string str1 没有加头文件<string>**