C++程序设计

主讲：王老师

尚德机构

学习是一种信仰

教材介绍

C++程序设计

（2008年版）

编著：刘振安

机械工业出版社

考试题型

单选题　　1分×20题 = 20分

填空题　　1分×20题 = 20分

程序改错题　4分×5题 = 20分

完成程序题　4分×5题 = 20分

程序分析题　5分×2题 = 10分

程序设计题 10分×1题 = 10分

第十章　面向对象设计实例

目录

第10章

面向对象

设计实例

§10.1 过程抽象和数据抽象

一、抽象的概念

抽象是从许多事物中舍弃个别的、非本质性特征，抽取共同及本质性特征的过程。

二、抽象的意义

1、尽管问题域中事物很复杂，但分析员不需了解和描述其全部特征，只需要分析研究与系统目标有关的事物及其本质特征，其余的都可以舍弃；

2、通过舍弃个体事物在细节上的差异，抽取其共同特征可以得到一些事物的抽象概念，如OOA（面向对象设计）中的类。

三、过程抽象和数据抽象

抽象是面向对象方法中使用最广泛的原则，早在面向对象方法出现之前就已经开始应用，主要是过程抽象和数据抽象。

1、**过程抽象**：是指任何一个完成确定功能的操作序列，其使用者都可以把它看作一个单一的实体；

2、**数据抽象**：是根据施加于数据之上的操作来定义数据类型，并限定数据的值只能由这些操作来修改和观察。

§10.2 发现对象并建立对象层

一、步骤

1、将问题域和系统责任作为出发点。前者侧重客观存在的事物与系统中对象的映射，后者侧重系统责任范围内的每一项职责都能落实到某些对象来完成，二者重合很多，但又不完全一致。

2、正确运用抽象原则。

OOA使用对象映射问题域中的事物。通过舍弃与当前目标与系统责任无关的事物及其特征来完成抽象。

3、寻找候选对象的基本方法。主要是从问题域、系统边界、系统责任三方面找出可能有的候选对象。

§10.2 发现对象并建立对象层

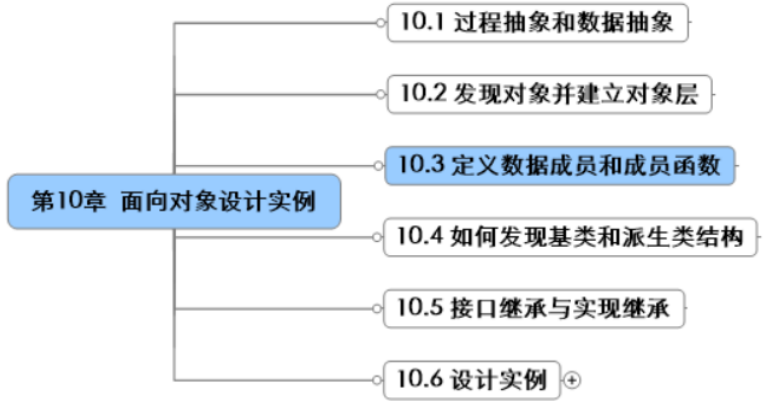
4、审查和筛选对象。原则为：

- 第一，舍弃无用对象，标准为看对象是否提供了有用的数据成员和成员函数；
- 第二，对象精简，即将只有一个数据成员和成员函数的对象合并到相关的对象中描述；
- 第三，舍弃目前不需要考虑的对象。

5、异常情况的检查和调整。包括以下的异常情况：

- 第一，类的成员不适合该类的全部对象；
- 第二，不同类的成员相同或相似；
- 第三，对同一事物的重复描述。

§10.3 定义数据成员和成员函数



数据成员包括静态和非静态两种。

为发现对象的数据成员，首先应借鉴以往的OOA成果，尽可能复用其中同类对象数据成员的定义，然后重点研究当前的问题域和系统责任，针对本系统找出每一类对象应有的数据成员。

一、寻找数据成员的一般方法

1、审查时去掉无用的数据。

2、确定在当前的问题域中，对象应有的数据成员。

3、根据系统责任的要求，考虑对象应有的数据成员。

4、考虑建立这个对象是为了保存哪些信息。

5、考虑为了在服务中实现其功能，需要增设哪些数据成员。

6、考虑对象有哪些需要区别的状态，是否需要增加一个数据成员来区分它们。

7、考虑用什么数据成员报市整体 - 部分结构合实例连接。

二、审查和筛选数据成员

通过数据成员是否满足下面条件来进行审查和筛选：

1、是否体现了以系统责任为目标的抽象。

2、是否描述了这个对象本身的特征。

3、是否符合人们日常的思维习惯。

4、是否可通过继承得到，基类中定义了的数据成员，都不要在派生类中出现。

5、是否可从其他数据成员直接导出。

三、定义成员函数

使用中要注意区分成员函数、非成员函数和友元函数三者。

设计一个类时，要注意接口功能不仅要完整紧凑，而且要使类的结构达到最小化。

§10.4 如何发现基类和派生类结构

一、学习当前领域的分类学知识

分析员应该学习一些与当前问题域有关的分类学知识，然后按照本领域已有的分类方法，可以找出一些与对象对应的基类与派生类。

二、按照常识考虑事物的分类

如问题域没有可供参考的现行分类方法，可按照自己的常识，从各种不同的角度考虑事物的分类，分类要考虑是否符合分类学常识，基类与派生类结构中的各个类之间的关系应符合分类学的常识和人类的日常思维方式。

§10.4 如何发现基类和派生类结构

三、构建基类与派生类

思路有两种：一种是把每一个类看作是一个对象集合，分析这些集合之间的包含关系，如果一个类是另一个类的子集，则它们就应组织到同一个基类与派生类结构中；另一种是看一个类是否具有另一个类的全部特征，又分两种情况：一是建立这些类时已经计划让某个类继承另一个类的全部成员，此时应该建立基类与派生类结构来落实这种继承；二是起初只是孤立的建立每一个类，现在发现一个类中定义的成员全部在另一个类中重新出现了，此时应考虑建立基类与派生类结构，把后者作为前者的派生类，以简化定义。

§10.4 如何发现基类和派生类结构

四、考察类的成员

从下面两方面考察每个类的成员：

一是看一个类的成员是否适合这个类的所有对象；

二是检查是否有两个或更多类含有一些共同的成员，如果有，则考虑若把这些共同的成员提取出来后能否构成一个在概念上是包含原来那些类的基类，组成一个基类与派生类结构。

§10.5 接口继承与实现继承

一、成员函数分类

类通过成员函数提供与外界的接口，成员函数分为实、虚、纯虚三种。

1、纯虚函数

纯虚函数最显著的两个特征是：它们必须由继承它的非抽象类重新说明；它们在抽象类中没有定义。

说明一个纯虚函数的目的是使派生类继承这个函数的接口，但这种抽象的基类从不给派生类提供实现。

§10.5 接口继承与实现继承

2、虚函数

虚函数的情况不同于纯虚函数，派生类继承虚函数的接口，虚函数一般只提供一个可供派生类选择的实现。之所以声明虚函数，目的在于使派生类既继承函数接口，也继承虚函数的实现。

§10.5 接口继承与实现继承

3、实函数

声明实函数的目的，是使派生类既继承这个函数的实现，也继承其函数接口。

4、应注意尽量避免重新定义继承的实函数

虽然在派生类中可以重定义基类的同名实函数，但是从使用安全角度来看，为了提高程序质量，在实际应用中应避免这样做。

二、函数接口的继承和函数实现的继承

假设有一个基类，其派生类通过公有继承方式，公有继承实际上分函数接口的继承及函数实现的继承两部分，可概括为：

1、继承的总是成员函数的接口，对于基类是正确的任何事情，对于它的派生类必须也是正确的，反之则不然；

2、声明纯虚函数的目的，是使派生类仅仅继承函数接口，而纯虚函数的实现则由派生类去完成；

3、声明虚函数的目的，是使派生类既能继承基类对此虚函数的实现，又能继承虚函数提供的接口；

4、声明实函数的目的，是使派生类既能继承基类对此实函数的实现，又能继承实函数提供的接口。

本章总结


第10章 面向对象设计实例
10.1 过程抽象和数据抽象
10.2 发现对象并建立对象层
10.3 定义数据成员和成员函数
10.4 如何发现基类和派生类结构
10.5 接口继承与实现继承
10.6 设计实例
10.6.1 使用包含设计的方法
10.6.2 使用包含的参考程序及运行结果
10.6.3 使用继承的设计方法
10.6.4 使用继承的参考程序和运行结果

1. 要求指针 p 既不可修改其本身的内容，也不可修改其所指向地址的内容，定义正确的是

A. const char * p = "ABCD";　　B. char * const p = "ABCD";

C. char const * p = "ABCD";　　D. const char * const p = "ABCD";

2. 在 C++ 中，类与类之间的继承关系具有

A. 自反性　　B. 对称性　　C. 传递性　　D. 反对称性

3. 下列关于对静态数据成员的描述中，正确的是

A. 静态数据成员不能用 public 控制符修饰

B. 静态数据成员可以直接用类名或者对象名来调用

C. 静态数据成员不可以被类的对象调用

D. 静态数据成员不能用 private 控制符修饰

4. 适宜采用 inline 定义函数情况是

A. 函数体含有循环语句　　B. 函数体含有递归语句

C. 函数代码多、不常调用　　D. 函数代码少、频繁调用

5. 通常拷贝构造函数的参数是

A. 某个对象的成员名　　B. 某个对象的指针名

C. 某个对象的引用名　　D. 某个对象名

6. 下列不是类的成员函数的是

A. 友元函数　　B. 构造函数　　C. 析构函数　　D. 拷贝构造函数

7. 类 Cat 是类 Animal 的公有派生类，类 Animal 和类 Cat 中都定义了虚函数 func()，p 是一个指向类 Animal 对象的指针，则 p -> Animal::func()将

A. 调用类 Animal 中的函数 func()

B. 调用类 Cat 中的函数 func()

C. 根据 p 所指的对象类型而确定调用类 Animal 中或类 Cat 中的函数 func()

D. 既调用类 Animal 中函数，也调用类 Cat 中的函数

8. 如果表达式 ++a 中的“ ++ ”是作为成员函数重载的运算符，若采用运算符函数调用格式，则可表示为

A. a. operator ++ (1)　　B. operator ++ (a)

C. operator ++ (a,1)　　D. a. operator ++ ( )

9. 实现两个相同类型数加法的函数模板的声明是

A. add(T x,T y)　　B. T add(x,y)

C. T add(T x,y)　　D. T add(T x,T y)

10. 以下基类中的成员函数表示纯虚函数的是

A. virtual void vf(int)　　B. void vf(int) =0

C. virtual void vf( ) =0　　D. virtual void yf(int){}

11. 假定一个类的构造函数为 A(int aa,int bb){a = aa ++ ;b = a * ++bb;},则执行 A x(4,5);语句后，x. a 和 x. b 的值分别为

A. 4 和 5　　B. 4 和 20　　C. 4 和 24　　D. 20 和 5

12. 下列运算符中，在 C++ 语言中不能重载的是

A. *　　B. >=　　C. ::　　D. /

13. 在编译指令中，宏定义使用哪个指令

A. #if　　B. #include　　C. #define　　D. #error

14. 以下能正确定义数组并正确赋初值的语句是

A. int N=5,b[N][N];

B. int a[2]={{1},{3},{4},{5}};

C. int c[2][]={{1,2},{3,4}};

D. int d[3][2]={{1,2},{3,4}};

15. 关于函数模板，描述错误的是

A. 函数模板必须由程序员实例化为可执行的函数模板；

B. 函数模板的实例化由编译器实现；

C. 一个类定义中，只要有一个函数模板，则这个类是类模板；

D. 类模板的成员函数都是函数模板，类模板实例化后，成员函数也随之实例化；

16. 按照标识符的要求，不能组成标识符的是

A. 连接符　　B. 下划线　　C. 大小写字母　　D. 数字字母

17. 已知类 A 是类 B 的友元，类 B 是类 C 的友元，则

A. 类 A 一定是类 C 的友元

B. 类 C 一定是类 A 的友元

C. 类 C 的成员函数可以访问类 B 的对象的任何成员

D. 类 A 的成员函数可以访问类 B 的对象的任何成员

18. 已知：p 是一个指向类 A 数据成员 m 的指针，A1 是类 A 的一个对象。如果要给 m 赋值为 5，正确的是

A. A1.p=5;　　B. A1->p=5;　　C. A1.*p=5;　　D. *A1.p=5;

19. 关于 this 指针的说法错误的是

A. this 指针必须显式声明

B. 当创建一个对象后，this 指针就指向该对象

C. 动态成员函数拥有 this 指针

D. 静态成员函数不拥有 this 指针

20. 已知:func( ) 函数是一个类的常成员函数,它无返回值,下列表示中,是正确的是

A. void func( ) const;　　B. const void func( );

C. void const func( );　　D. void func(const);

21. _____通常使用对象的引用来初始化创建中的对象。

22. 抽象类中至少要有一个_____函数。

23. vector 类中用于删除向量中的所有对象的函数是_____。

24. 重载的运算符保持其原有的操作数、_____和结合性不变。

25. 执行下列代码

```
string str("NihaoC++");
cout << str. substr(5,3);
```

程序的输出结果是_____。

26. C++ 中有两种继承：单一继承和_____。

27. C++ 支持两种多态性：_____时的多态性和运行时的多态性。

28. 在用 class 定义一个类时，数据成员和成员函数的默认访问权限是_____。

29. C++ 流库预定义了 4 个流，它们分别是 cin、cout、clog 和_____。

30. 在 C++ 中，有两种给出注释的方法。一种是沿用 C 语言的注释符，即 /**/。另一种是从它开始，直到它所在行尾的字符都为注释的注释符，即_____。

31. 使用 new 为 int 数组动态分配 10 个存储空间是_____。

32. 假设 int a = 1, b = 2;则表达式(++ a/b) * b — 的值为_____。

33. C++ 程序的源文件扩展名为_____。

34. 语句序列

```
ifstream infile;
infile. open("data. dat");
```

的功能可用一个语句实现,这个语句是_____。

35. 已知有 20 个元素 int 类型向量 V1,若用 V1 初始化为 V2 向量,语句是_____。

36. 为了避免可能出现的歧义，C++ 对 if…else 语句配对规则规定为：else 总是与_____配对。

37. C++ 程序的编译是以_____为单位进行的。

38. 重载函数在参数类型或参数个数上不同，但_____必须相同。

39. 举出 C++ 中两种代码复用的方式：_____、复用。

40. 下面程序运行的结果是_____。

```
#include <iostream.h>
void main()
{
cout.fill('*');
cout.width(6);
cout << "hello" << endl;
cout.clear();
```

程序改错题:

```
#include <iostream>
#include <string>
using namespace std;
class Base
{
public:
Base(const char *na){strcpy(name,na);}
private:
char name[80];
};
```

```
class Derived:public Base
{ public:
Derived(const char *nm):Base(nm){}
void show();
};
void Derived::show()
{ cout<<"name:"<<name<<endl;
}
void main()
{ Derived b1("B");
b1.show();
}
```

程序改错题:

```
#include <iostream>
using namespace std;
void setzero(int &a)
{
    a = 0;
}
int main()
{
    int x1 = 10; x2 = 20;
    setzero(x1);
    setzero(x2);
    cout << x1 << x2 << endl;
    return 0;
}
```

43.
```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int *p1 = NULL;
    int *p2 = a;
    cout << p1 << *p2 << endl;
    return 0;
}
```

程序改错题:

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

程序改错题：

```
#include <iostream>
using namespace std;
class INTEGER
{
public:
    INTEGER(int a){this->a=a;}
protected:
    int a;
    friend void Print(const INTEGER& obj);
};
void INTEGER::Print(const INTEGER & obj)
{
    cout<<obj.a;
}
```

```
void main()
{
    INTEGER obj(4);
    Print(obj);
}
```

完成程序题：

完成下面类中成员函数的定义。

```
#include <iostream>
#include <string>
using namespace std;
class str
{private:
char *st;
public:
    str(char *a)
    {
        set(a);
    }
    _____//运算符"="重载函数
    {
        delete st;
        set(a.st);
        return *this;
    }
    void show(){cout<<st<<endl;}
    ~str(){delete st;}
    void set(char *s)
    {
        _____
        strcpy(st,s);
    }
};
void main()
{
    str s1("hello"),s2("world");
    s1.show(),s2.show();
    s2=s1;
    s1.show(),s2.show();
}
```

完成程序题：

Test_Static 类内有一静态成员变量 num，且初始值设为 15。

```
#include <iostream>
using namespace std;
class Test_Static
{
private:
    ____________________
public:
    Test_Static(int);
    void print();
};
____________________
Test_Static::Test_Static(int n)
{
    num = n;
}
```

```
void Test_Static::print()
{
    cout << num << endl;
}
void main()
{
    Test_Static ts(10);
    ts.print();
}
```

完成程序题：

任意输入 10 个同学的成绩，计算其平均成绩。要求用函数 average( )计算平均成绩。主函数输入数据并输出结果。

```
#include" stdafx. h"
#include < iostream >
using namespace std;
void main( void)
{
    float average( float a[ ]) ;
    float score[ 10] ;
    for( _____ )
    {
        cin >> score[ i] ;
    }
    cout << " average: " << average( score) << endl;
}
```

```
float average( float a[ ])
{
    float sum =0;
    for( int i =0;i < 10;i ++ )
    {
        _____;
    }
    return ( sum/10) ;
}
```

完成程序题：

下面程序中 Base 是抽象类。请在下面程序的横线处填上适当内容，以使程序完整，并使程序的输出为：

```
Der1 called!
Der2 called!
```

```
#include <iostream.h>
class Base
{public:
_____;
};
class Der1:public Base
{public:
void display(){cout<<"Der1 called!"<<endl;}
};
class Der2:public Base
{public:
void display(){cout<<"Der2 called!"<<endl;}
};
void fun(_____)
{p->display();}
void main()
{Der1 b1;
Der2 b2;
Base *p=&b1;
fun(p);
p=&b2;
fun(p);
}
```

完成程序题：

```
#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] <<" ";
}
```

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();
}
```

程序分析题：

```
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;
class student
{
    char name[8];
    int deg;
    char level[7];
    friend class process; // 说明友元类
public:
    student(char na[],int d)
    {
        strcpy(name,na);
        deg=d;
    }
};
class process
{
public:
    void trans(student &s)
    {
        int i=s.deg/10;
        switch(i)
        {
        case 9:
            strcpy(s.level,"优");break;
        case 8:
            strcpy(s.level,"良");break;
        case 7:
            strcpy(s.level,"中");break;
        case 6:
            strcpy(s.level,"及格");break;
        default:
            strcpy(s.level,"不及格");
        }
    }
    void show(student &s)
    {
        cout<<setw(10)<<s.name<<setw(4)<<s.deg<<setw(8)<<s.level<<endl;
    }
};
void main()
{
    student st[]={student("Jack",78),student("Lucy",92),
        student("Lily",62),student("Tom",99)};
    process p;
    cout<<"结果:"<<"姓名"<<setw(6)<<"成绩"<<setw(8)<<"等级"<<endl;
    for(int i=0;i<4;i++)
    {
        p.trans(st[i]);
        p.show(st[i]);
    }
}
```

程序设计题：

六、程序设计题(本大题共1小题,共10分)

53. 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,主函数已给出,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;
}
```

```
#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;
}
```

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;
};
```

```
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;
}
```

```
#include <iostream>
using namespace std;
class A{
private:
    int val;
public:
    A(int i){val=i;}    //含一个整型参数的类A的构造函数
    int value(int a){return val+a;}
};
void main( ){
    int(A::*pfun)(int);    //声明指向类A的成员函数的指针pfun
    pfun=A::value;    //指针指向A的具体的成员函数value
    A obj(10);    //创建类A的对象obj，并调用构造函数初始化
    cout<<(obj.*pfun)(15)<<endl;  //对象obj调用指针指向的成员函数value，输出25
    A *pc=&obj;    //创建类型为A的指针pc，该指针指向类A的对象obj
    cout<<(pc->*pfun)(15)<<endl;  //指针pc调用成员函数value，输出25
}
```

25
25

三种继承方式下基类成员在派生类中的访问属性

基类私有成员	继承方式	基类成员在派生类中的访问属性
Private	Public	不可直接访问
Private	Protected	不可直接访问
Private	Private	不可直接访问
Protected	Public	Protected
Protected	Protected	Protected
Protected	Private	Private
Public	Public	Public
Public	Protected	Protected
Public	Private	Private

访问属性	作用
private	只允许该类的成员函数及友元函数访问，不能被其他函数访问
protected	既允许该类的成员函数及友元函数访问，也允许其派生类的成员函数访问
public	既允许该类的成员函数访问，也允许类外部的其他函数访问

```
#include <iostream>
using namespace std;
class Point{
private:
        int x,y;
public:
        Point(int a,int b){x=a;y=b;cout<<"点"<<" ";}
        void Showxy( ){cout<<"x="<<x<<",y="<<y<<" ";}
        ~Point( ){cout<<"删除点"<<" ";}
};
class Rectangle:public Point{
private:
        int H,W;
public:
        Rectangle(int a,int b,int h,int w):Point(a,b){H=h;W=w;cout<<"矩形"<<" ";} //构造函数初始化列表
        void Show( ){cout<<"H="<<H<<",W="<<W<<" ";}
        ~Rectangle( ){cout<<"删除矩形"<<" ";}
};
void main(){
        Rectangle r1(3,4,5,6);    //生成派生类对象r1，r1先后调用基类和派生类的构造函数进行初始化
        r1.Showxy( );    //派生对象调用基类的成员函数
        r1.Show( );    //派生对象调用派生类的成员函数
}
```

点 矩形 x=3,y=4 H=5,W=6 删除矩形 删除点

```
#include <iostream>
using namespace std;
class A{
private:
        int a;
public:
        A(){ cout<<"construct A"<<endl;}
        void setA(int x){a=x;}
        void showA( ){cout<<"a="<<a<<endl;}
};
class B{
private:
        int b;
public:
        B(){ cout<<"construct B"<<endl;}
        void setB(int x){b=x;}
        void showB( ){cout<<"b="<<b<<endl;}
};
class C:public B,public A{
private:
        int c;
         A  a; B b;
public:
        C(){ cout<<"construct C"<<endl;}
        void setC(int x,int y){c=x;setB(y);}
        void showC( ){showB( );cout<<"c="<<c<<endl;}
};
```

```
void main( ){
        C obj;
        obj.setA(53);
        obj.showA( );    //输出信息"a = 53"
        obj.setC(55,58);
        obj.showC( );    //输出信息"b = 58 c = 55"
}
```

```
construct B
construct A
construct A
construct B
construct C
a=53
b=58
c=55
```

1．一个函数的功能不太复杂，但要求被频繁调用，选用最适合的是

A．内联函数　　B．重载函数　　C．递归函数　　D．嵌套函数

2．假定有类AB，有相应的构造函数定义，能正确执行”AB a(4)，b(5)，c[3]，*p[2]={&a，&b}；”语句，请问执行完此语句后共调用该类析构函数的次数为

A．14　　B．5　　C．3　　D．1

3．在下面有关析构函数特征的描述中，正确的是

A. 一个类中可以定义多个析构函数　　B．析构函数名与类名完全相同

C．析构函数不能指定返回类型　　D．析构函数可以有一个或多个参数

4．派生类的对象对它的哪一类基类成员是可以访问的

A．公有继承的基类的公有成员　　B. 公有继承的基类的保护成员

C．公有继承的基类的私有成员　　D. 保护继承的基类的公有成员

5．如果A是抽象类，刚下面正确的是

A. A中没有纯虚函数　B．A a;　　C．A a[3];　　D．A * pa;

6．下列关于纯虚函数与抽象类的描述中，错误的是

A．纯虚函数是一种特殊的函数，它允许没有具体的实现

B．抽象类是指具有纯虚函数的类

c．一个基类的说明中有纯虚函数，该基类的派生类一定不再是抽象类

D．抽象类只能作为基类来使用，其纯虚函数的实现由派生类给出

7．cout是C++的

A．关键字　　B．对象　　C．类　　D．运算符

8．下列哪个函数不是类的成员函数

A．构造函数　　B．析构函数　　C．友元函数　　D．拷贝构造函数

9．下列有关重载函数的说法中错误的是

A．重载函数必须具有不同的返回值类型

B． 重载函数名必须相同

C．重载丞数参数个数可以不同

D．重载函数必须有不同的形参列表

10. 下面说法正确的是

A．生成对象时调用析构函数

B．定义类时必须写出该类的构造函数

C．调用构造函数时必须在主函数中明确写出调用格式

D．析构函数不可以重载

11．在编译指令中，宏定义使用指令

A．#define　　B．#include　　C．#typedef　　D．#friend

12．假设声明了以下的西数模板，错误的调用语句是

```
template <class T>
T max(T x,T y) { return (x>y)? x:y;}
并定义了 int i;char c;
```

A. max(i,i)　　B. max(c,c)　　C. max((int)c,i)　　D. max(i,c)

13. 若有说明：int n=2，*p=&n，*q=p；，则以下非法的赋值语句是

A. n=*q　　B． p =n　　C． p =q　　D．*q=* p

15．类模板template<class T>class?({...}；，其中友元函数f对特定类型T(如int)，使函数f(X<int>&)成为X<hat>模板类的友元，则其说明应为

A．friend void “)；　　B．friend void f(X<T>&)；

C．friend void A::f()；　　D．friend void C(T)；

16．动态编联所支持的多态性称为运行时的多态性，支持的函数是

A．构造函数　　B．友元函数　　C．继承　　D．虚函数

17．用运算符delete删除一个动态对象时

A．首先为该动态对象调用构造函数，再释放其占用的内存

B．首先释放该动态对象占用的内存，再为其调用构造函数

C．首先为该动态对象调用析构函数，再释放其占用的内存

D．首先释放该动态对象占用的内存，再为其调用析构函数

18．下列字符常量的写法中，错误的是

A．′\t′　　B．′b′　　C．′*′　　D．′a′

19．C++类体系中，不能被派生类继承的有

A. 静态数据成员　　B. 构造函数　　C．虚函数　　D．静态成员函数

20．下列对派生类的描述中，错误的是

A．派生类至少应有一个基类

B．派生类的成员除了自己定义的成员外，还包含了它的基类成员

C．基类中成员访问权限继承到派生类中都保持不变

D．一个派生类可以作为另一个派生类的基类

21．C++注释方式"／／"的有效范围从"／／"至_____结束。

22．C++程序有且只能有一个名为_____的主函数。

23．语句"_____；"用来向屏幕输出显示信息"Hello!"。

24．所谓"_____"就是将一个新标识符和一块已经存在的存储区域相关联。

25．动态分配内存使用关键字_____，释放内存使用关键字delete。

26．表达式"20 / 3*sqrt(4．0) / 5"值的数据类型是_____。

27．面向对象的程序设计方法是以_____代表求解问题的中心环节。

28．对象的——只能由这个对象的操作来存取。

29．使用关键字inline说明的函数称为_____函数。

30．类对象一般都包括数据成员和_____。

31．将对象作为函数参数，是将实参对象的值传递给_____。

32．C++函数的返回值类型可以是除_____和函数以外的任何类型。

33. 函数重载可使一个函数名具有多种功能，称这种特性为_____。

34．_____是类的实例。

35．关键字private、public和_____以后的成员分别叫做私有成员、公有成员和保护成员。

36. 使用关键字const修饰的数据成员称为_____数据成员。

37．对MyFirst类定义析构函数是_____。

38．insert(iterator it，const T&)是向it所指向量位置前_____个对象。

39．说明纯虚函数的一般形式为”class类名{_____函数类型函数名(参数列表)=0；}；”。

40．运算符重载需要使用关键字”_____”。

祝大家顺利通过考试！