

## 一、选择题

- 1、关于封装，下列说法中不正确的是 ( A )
  - (A) 通过封装，对象的全部属性和操作结合在一起，形成一个整体
  - (B) 通过封装，一个对象的实现细节被尽可能地隐藏起来 (不可见)
  - (C) 通过封装，每个对象都成为相对独立的实体
  - (D) 通过封装，对象的属性都是不可见的
- 2、假定 AA 为一个类，int a()为该类的一个成员函数，若该成员函数在类体外定义，则函数头为 ( A )
  - (A) int AA::a ( )
  - (B) int AA:a ( )
  - (C) AA::a ( )
  - (D) AA::int a ( )
- 3、一个类中是否可以允许一个无参构造函数和一个全部都是默认参数的构造函数共存 ( A )
  - (A) 不允许
  - (B) 允许
  - (C) 无法确定
  - (D) 视构造函数确定
- 4、在 C++ 中，cin 是个 ( B )
  - (A) 类
  - (B) 对象
  - (C) 模板
  - (D) 函数
- 5、析构函数的作用是 ( D )
  - (A) 一般成员函数的初始化
  - (B) 类的初始化
  - (C) 对象的初始化
  - (D) 在删除类创建的对象前完成一些清理工作
- 6、假定一个类 AB 含有一个浮点型数据成员 a，当用户不定义任何构造函数时，系统为该类型定义的无参构造函数为 ( D )
  - (A) AB() {a=0;}
  - (B) AB(float aa=0): a(aa) {}
  - (C) A(){}
  - (D) AB() {}
- 7、C++ 语言是由 C 语言逐渐发展演变而来的。与 C 语言相比，它在求解问题方法上进行的最大改进是 ( A )
  - (A) 面向过程
  - (B) 面向对象
  - (C) 安全性
  - (D) 复用性
- 8、C++ 的源文件的扩展名为 ( A )
  - (A) .cpp
  - (B) .exe
  - (C) .obj
  - (D) .lik
- 9、下列说法中，不是构造函数的特征是 ( D )
  - (A) 构造函数的函数名与类名相同
  - (B) 构造函数可以重载
  - (C) 构造函数可以设置缺省参数
  - (D) 构造函数必须指定类型说明
- 10、函数重载是指 ( A )
  - (A) 两个或两个以上的函数取相同的函数名，但形参的个数或者类型不同
  - (B) 两个以上的函数取相同的名字和具有相同的参数个数，但形参的类型可以不同
  - (C) 两个以上的函数名字不同，但形参的个数或者类型相同

(D) 两个以上的函数取相同的函数名, 并且函数的返回类型相同

11、有如下类定义:

```
class MyClass {
```

```
    int Value;
```

```
Public:
```

```
    MyClass(int n) : Value(n) {} //如果没有,自动生成默认
```

```
                                //copy 构造函数
```

```
                                //赋值构造函数
```

```
    int GetValue() {return Value;} };
```

则类 MyClass 中, 构造函数的个数是 ( C )

(A) 1个 (B) 2个

(C) 3个 (D) 4个

12、面向对象程序设计的三大特性不包括() D

A:封装性 B:继承性 C:多态性 D:健壮性

13、在 C++ 语言中,数据封装要解决的问题是() D

A:数据的规范化 B:便于数据转换

C:避免数据丢失 D:防止不同模块之间数据的非法访问

14、下列各项中符合函数重载必须满足的条件的是 D

A:必须有不同的参数个数 B:对应的参数类型必须不相同

C:A 和 B 必须同时满足 D:A 和 B 只要满足一个即可

15、下列存储类标识符中,要求通过函数来实现一种不太复杂的功能,并且要求加快执行速度,选用( B )合适。

A:内联函数; B:重载函数; C:递归调用; D:嵌套调用。

16、下列关于类定义的说法中,正确的是()

A:类定义中包括数据成员和函数成员的声明

B:类成员的缺省访问权限是保护的

C:数据成员必须被声明为私有的

D:成员函数只能在类体外进行定义

17、四组函数原型说明中,满足 C++ 函数重载规则的是 C

~~A: float f(float x); void f(float y);~~

~~B: float f(float x); void f2(float x, float y);~~

C: float f(float x, int y); void f(float y);

~~D: float f(float y); float f(float y);~~

18、已知类 A 中的一个成员函数的说明如下:

void Set(A &a); 则该函数的参数 "A &a" 的含义是

A:指向 A 的指针为 a

B:将变量 a 的地址赋给类 A

C:类 A 对象引用 a 用作函数的形参

D:变量 A 与 a 按位与后作函数参数

19、类定义 class A { int x; static int y; }; 问 32 机中 sizeof A 等于 A

A:4 B:8 C:0 D:不定

20、自定义类型中的哪种成员的访问不受限制(A)

A:public B:protected C:private D:以上都不是

21、在 C++ 中,编写一个内联函数 Fun,使用 int 类型的参数,求其平方并返回,返回值也为 int



类型,下列定义正确的是 (B)

A: int Fun(int x){return x\*x;}

B: inline int Fun(int x){return x\*x;}

C: int inline Fun(int x){return x\*x;}

D: int Fun(int x){inline return x\*x;}

22、类定义 class A{int x;static int y;},问 32 机中 sizeof A 等于 ( ) A

A:4 B:8 C:0 D:不定

23、指出下列输出语句哪些是不合法的 ( ) D

A: std::cout << "/\*"; B: std::cout << "\*/";

C: std::cout << "/\*\*/"; D: std::cout << /\* "\*/" \*/;

24、一个函数调用语句为 fun((e1,e2,e3),(e4,e5),e6);,其实参个数为 A

A:3 B:4 C:5 D:6

25、所谓数据封装就是将一组数据和与这组数据有关操作组装在一起,形成一个实体,这实体也就是 A

A:类 B:对象 C:函数体 D:数据块

26、面向对象程序设计中的数据隐藏指的是 ( ) D

A:输入数据必须输入保密口令 B:数据经过加密处理

C:对象内部数据结构上建有防火墙

D:对象内部数据结构的不可访问性

27、存储以下数据,占用存储字节最少的是 (B).

A:0 B:'0' C:"0" D:0.0 → float 型, 4个字节

28、面向对象程序设计将数据与 ( ) 放在一起,作为一个相互依存、不可分割的整体来处理 A

A:对数据的操作 B:信息 C:数据隐藏 D:数据抽象

29、下列是模板声明的开始部分,其中正确的是 ( )

A: template<T> B: template<class T1,T2>

C: template<class T1, class T2>

D: template<class T1;class T2>

30、类中定义的成员默认为 ( ) 访问属性. B

A: public B: private C: protected D: friend

31、使用输入输出操作符 setw,可以控制 B. set width

A:输出精度 B:输出宽度 C:对齐方式 D:填充字符

32、面向对象系统的 ( ) 是一种信息隐藏技术,目的在于将对象的使用者与设计者分开,使用者不必知道对象行为实现的细节,只需用设计者提供的协议命令对象去做即可 B

A:多态性 B:封装性 C:继承性 D:复用性

33、关于 this 指针使用说法正确的是 (A)

(A) 保证每个对象拥有自己的数据成员,但共享处理这些数据的代码

(B) 保证基类私有成员在子类中可以被访问

(C) 保证基类保护成员在子类中可以被访问

(D) 保证基类公有成员在子类中可以被访问

34、构造函数是在 (B) 时被执行的.

A:程序编译 B:创建对象 C:创建类 D:程序装入内存

35、下面程序运行时的输出结果是 ( )

```
#include<iostream>
```

```
using namespace std;
```

```

class test {
private:
    int a;
public:
    test(){cout<<"constructor"<<endl;}
    test(int a){cout<<a<<endl;}
    test(const test &test){
        a=test.a;
        cout<<"copy constructor"<<endl;}
    ~test(){cout<<"destructor"<<endl;}
};
int main()
{
    test A(3);
    return 0;
}

```

子对象构造 → 当前构造 → 当前析构  
→ 子对象析构

A:3  
B:constructor  
destruclor  
C:copy constructor  
destructor  
destructor  
D:3  
destruclor

36、假定 AB 为一个类,则( )为该类的拷贝构造函数的原型说明

A:AB(AB x); B:AB(int x); C:AB(const AB& x)

D:void AB(AB& x);

37、拷贝构造函数的作用是 C

A:进行数据类型的转换 B:用对象调用成员函数

C:用对象初始化对象 D:用一般类型的数据初始化对象

38、下面的程序执行时,MyClass 类的拷贝构造函数被调用的次数是

```

class MyClass {
public:
    MyClass(int n){number = n;} //拷贝构造函数
    MyClass(const MyClass &other){    number=other.number;}
    ~MyClass(){}
private:
    int number;
};
MyClass fun(MyClass p) {
    MyClass temp(p);    return temp;}
int main() {
    MyClass obj1(10), obj2(0);
}

```

```

MyClass obj3(obj1);
obj2=fun(obj3);
return 0;
}

```

A:5      B:4      C:3      D:2

39、要与 new 配合使用的是 ( A )

A:delete      B:free      C:delete[]      D:destroy

40、假设 Sample 是个类,则语句"Sample a[2],\*p[3];"调用构造函数的次数为( ) C

A:0      B:2      C:3      D:5

41、通常,拷贝构造函数的形参类型是 ( A )

A:某个对象类型      B:某个对象的成员类型

C:某个对象的引用类型      D:某个对象的指针名

42、关于默认构造说法错误的是 ( B )

A:默认构造函数,调用时可以不提供参数

B:没有默认构造不能定义数组

✓ C:一个类必须存在默认构造函数

D:默认构造函数可能由系统提供

43、class MyClass {

private: int id; char gender; char \*phone;

public:

MyClass():id(0),gender('#'),phone(NULL) { }

MyClass(int no, char ge='#', char \*ph= NULL)

{ id=no;gende=ge;phone=ph; }

};

下列类对象定义语句中错误的是 ( )

A:MyClass myObj;

B:MyClass myObj(11, "13301111155");

C:MyClass myObj(12, 'm');

D:MyClass myObj(12);

44、声明函数 int Func(int a=0,int b=3,int c=4)相当于声明了几个函数形式( )

A:1

B:2

C:3

D:4

45、有以下类定义 class Point { public:

Point(int x=0, int y=0){ \_x = x; \_y = y; }

void Move(int xOff,int yOff){\_x+=xOff;\_y+= yOff;}

void Print() const

{cout<< '(' <<\_x<<','<<\_y<<')'<<endl;}

private: int \_x, \_y; };

下列语句中会发生编译错误的是

A:Point pt; pt.Print();

B:const Point pt;pt.Print();

C:Point pt; pt.Move(1, 2);

D:const Point pt; pt.Move(1, 2);



## 二、填空

1、

```
#include <iostream.h>
class base1 {
public:
base1() { cout << "Constructing base1\n"; }
~base1() { cout << "Destructing base1\n"; }
};
class base2 {
public:
base2() { cout << "Constructing base2\n"; }
~base2() { cout << "Destructing base2\n"; }
};
class derived: public base1, public base2 {
public:
derived() { cout << "Constructing derived\n"; }
~derived() { cout << "Destructing derived\n"; }
};
int main()
{
derived ob;
return 0;}
```

运行结果:

```
Constructing base1
Constructing base2
Constructing derived
Destructing derived
Destructing base2
Destructing base1
```

2、 class B{

```
public:
    B(int a){cout << a << endl;}
    ~B(int){cout << "~B" << endl;}
};
```

class A    A 依赖于 B

```
{
    const float a; //声明类时,不允许给变量赋值.
                    //常量要求必须赋初值.
    int b;
    int &c;
    B d;    //d 是 A 的子对象
```

```

public:
    A(int ):d(1),a(3.14),c(b){
        cout << "A" << endl;
    }
    ~A(){cout << "~A" endl;}

};

```

```

int main()
{
    A a;
    return 0;
}

```

运行结果:

```

B
A
~A
~B

```

3、假定 class B 为一个类，b 为该类的公有数据成员，show()为该类公有函数成员，x 为该类的一个对象，则访问 x 对象中数据成员 b 的格式为\_\_x.b\_\_，访问 x 对象中函数成员 show()的格式为\_\_x.show()\_\_。

4、类的定义中，既包含数据成员，也包含\_\_函数\_\_成员。

5、若在类 A 的定义体中只给出了一个成员函数的原型，则在类外给出完整定义时，其函数名前必须加上\_\_类名::\_\_。

6、当执行 cin 语句时，从键盘上输入每个数据后必须接着输入一个\_>>\_符，然后才能继续输入下一个数据。

7、用于输出表达式值的 cout 是\_\_输出流对象\_\_。

8、C++语言中用于引用一个对象的符号是\_\_&\_\_。

9、将指向对象的指针作为函数参数，形参是对象指针，实参是对象的\_\_地址\_\_。

10、若要使用 C++标准类库，需要指定相应的命名空间，所使用的语句是\_using namespace std\_。

11、面向对象的程序设计具有的主要基本特征：\_\_封装\_\_、\_\_继承\_\_、\_\_多态\_\_。

12、类的定义中，既包含数据成员，也包含\_\_函数\_\_成员。

13、C++中默认 main 函数的返回类型是\_\_int\_\_，所以在定义 main 函数时可以不写返回值类型。

14、定义函数时,在函数的类型前加上修饰词\_\_inline\_\_，指明将该函数定义为内联函数。

15、在调用函数 f1()的过程中,又调用了函数 f1(),这种调用称为\_\_递归\_\_调用。

16、去除构造函数隐式类型转换功能的关键字是\_\_explicit\_\_

```

17、class A{
    float x,y;
public:
    A(float a, float b=10) { x=a; y=b; }
    A() { x=0; y=0;}
    void Print(void) { cout<<x<<'\\t'<<y<<endl; }
}

```

```
};
void main(void)
{   A  a1, a2(20), a3(3.0, 7.0);
    a1.Print();
    a2.Print();
    a3.Print();
}
```

输出结果:

```
0      0
20     10
3      7
```

```
18、 class  A{
    float  x,y;
public:
    A(float a, float b){x=a;y=b;cout<<"初始化自动局部对象\n";}
    A(){ x=0; y=0; cout<<"初始化静态局部对象\n";}
    A(float a){ x=a; y=0; cout<<"初始化全局对象\n";}
    void Print(void){ cout<<x<<"\t"<<y<<endl; }
};
```

```
A  a0(100.0);//定义全局对象
void f(void)
{ cout<<" 进入 f()函数\n";
  A  a2(1,2);
  static  A  a3; //初始化局部静态对象 }
void main(void)
{   cout<<"进入 main 函数\n";
    A  a1(3.0, 7.0);//定义自动局部对象
    f();    f();  }
```

输出结果:

```
初始化全局对象
进入 main 函数
初始化自动局部对象
进入 f()函数
初始化自动局部对象
初始化静态局部变量
进入 f()函数
初始化自动局部对象
```

```
19、 class  A{
    float  x,y;
public:
    A(float a, float b)  {   x=a;y=b;}
    A() {   x=0; y=0;  }
    void Print(void){ cout<<x<<"\t"<<y<<endl; }
};
```



```
void main(void)
{   A   *pa1,*pa2;
    pa1=new   A(3.0, 5.0);//用 new 动态开辟对象空间，初始化
    pa2=new A;//用 new 动态开辟空间，调用构造函数初始化
    pa1->Print();
    pa2->Print();
    delete   pa1; //用 delete 释放空间
    delete   pa2; //用 delete 释放空间
}
```

输出结果：

```
3   5
0   0
```

### 三、简答

1、构造和析构的相同点和不同点：

相同点：无返回值\函数名\自动调用；

不同点：一个类可以有多个构造，但有且仅有一个析构，构造可以有参数,也可以没有，析构没有。

2、重载的条件：参数的个数 或者 类型 至少有一个不一样。