



# 厦门大学《C++程序设计》期中考试

姓名：\_\_\_\_\_ 学号：\_\_\_\_\_

得分	一	二	三	四	总分	
----	---	---	---	---	----	--

## 一. 单选题

- 编写 C++ 程序一般需经过的几个步骤依次是 ( )。  
A. 编辑、编译、调试、运行  
B. 编译、编辑、连接、运行  
C. 编译、调试、连接、运行  
D. 编辑、编译、连接、运行
- 下列属于构造数据类型的有 ( )。  
①指针类型 ②枚举类型 ③引用类型 ④字符类型 ⑤逻辑类型 ⑥类  
A. ①⑤⑥  
B. ②③④  
C. ①②③  
D. ④⑤⑥
- 关于函数说法正确的是 ( )  
A. 在 C++ 的函数内部可以定义形参、局部常量、局部变量、函数。  
B. 在不同的源文件中，对同一个函数的声明只能对它的每一个参数指定一次默认值。  
C. 函数的调用不可以嵌套。  
D. 在函数体中不能用 goto 语句转向函数外。
- 下列说法错误的是 ( )  
A. const 成员函数只能获取对象的状态，而不能改变对象的状态。  
B. 在编译指令中，宏定义使用的指令是 #define。  
C. 在一个命名空间中定义的全局标识符，其作用域为该命名空间。  
D. 解决对小函数频繁调用效率不高问题的机制为重载函数。
- 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 ( ) 合适。  
A. 嵌套调用      B. 内联函数      C. 递归调用      D. 重载函数
- 下列有关重载函数的说法中正确的是 ( )  
A. 重载函数必须有不同的形参列表。  
B. 重载函数参数个数必须相同。  
C. 重载函数必须具有不同的返回值类型。  
D. 重载函数名可以不同。

7. 关于 new 运算符的下列描述中，错误的是（ ）。
- A. 它可以用来动态创建对象和对象数组。
  - B. 使用它创建对象时要调用构造函数。
  - C. 使用它创建对象数组时必须指定初始值。
  - D. 使用它创建的对象或对象数组可以使用运算符 delete 删除。
8. 以下正确地定义函数指针的是（ ）。
- A. double (\*fp) (int)
  - B. double fp\* (int)
  - C. double \*fp (int)
  - D. double (fp) (\*int)
9. 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，也就是（ ）。
- A. 数据块
  - B. 类
  - C. 函数体
  - D. 对象
10. 下列中，类 A 的拷贝构造函数的定义正确的是（ ）。
- A. A()
  - B. A(A& a)
  - C. A(A\* a)
  - D. A(A a)
11. class A
- ```

{
    int x, y;
    public:
        A() { x = y = 1; }
        void f() { x++; y-- }
};
class B
{
    int z;
    A a;
    public:
        B() { z = 1; }
        B(const B& b) { z = b.z; }
        void f() { z++; a.f(); }
};
.....
B b1;
b1.f();
B b2(b1);

```
- 上述代码执行后，b2.z, b2.x 和 b2.y 的值分别是（ ）。
- A. 2, 1, 1
  - B. 2, 2, 0
  - C. 1,1,1
  - D. 1, 2, 0
12. 下列关于析构函数的描述中，错误的是（ ）。
- A. 析构函数可以重载。
  - B. 析构函数无返回类型。
  - C. 一个类只有一个析构函数。
  - D. 注销对象时，系统自动调用析构函数。

13. 静态成员函数没有 ( )。  
A. 返回值    B. 返回类型    C. this 指针    D. 指针参数
14. 下列关于重载运算符的说法正确的是 ( )  
A. 单目操作符可以被重载成双目操作符。  
B. 运算符 “.” 不能被重载。  
C. 运算符 “->” 不能被重载。  
D. 运算符 “[]” 不能被重载。
15. 若对类 Value 中重载的类型转换运算符 int 进行声明, 则正确的是 ( )。  
A. int operator int ();  
B. int operator int (Value);  
C. operator int ();  
D. operator int (Value);
16. 如果从原有类定义新类可以实现的是 ( )。  
A. 信息隐藏    B. 数据封装    C. 数据抽象    D. 继承机制
17. C++中, 不能被派生类继承的有 ( )。  
A. 构造函数    B. 成员函数    C. 静态成员函数    D. 虚函数
18. class Base  
{  
    private:  
        void fun1() const { cout<<"fun1"; }  
    protected:  
        void fun2() const { cout<<"fun2"; }  
    public:  
        void fun3() const { cout<<"fun3"; }  
};  
class Derived: protected Base  
{  
    public:  
        void fun4() const { cout<<"fun4"; }  
};  
int main() {  
    Derived obj;  
    obj.fun1();    // (1)  
    obj.fun2();    // (2)  
    obj.fun3();    // (3)  
    obj.fun4();    // (4)  
    return 0;  
}  
没有语法错误的是 ( )。  
A. (1)    B. (2)    C. (3)    D. (4)

## 二. 填空题

19. 分别使用引用类型和指针类型，交换两个 `int` 型变量的值。

```
void swap (      ,      ) //使用引用类型
{
    int tmp;
```

```

}
```

```
void swap (      ,      ) //使用指针类型
{
    int tmp;
```

```

}
```

20. 实现前置和后置自增操作符`++`的重载。

```
class Point
{
    public:
        Point(int x, int y){ X=x; Y=y; }
        ~Point(){}
        Point& operator ++ () //前置自增操作符
        {

        }

        const Point operator ++ (    ) //后置自增操作符
        {

        }

    private:
        int X,Y;
};
```

21. 下面的指针 `p` 分别指向什么。

|                               |       |
|-------------------------------|-------|
| (1) <code>int *p;</code>      | _____ |
| (2) <code>int *p[3];</code>   | _____ |
| (3) <code>int (*p)[3];</code> | _____ |
| (4) <code>int (*p)();</code>  | _____ |

22. 请写出下面程序的运行结果：（本题 4 分）

```
class A
{ public:
    A() { cout << "A's default constructor" << endl; }
    A(const A&) { cout << "A's copy constructor" << endl; }
    ~A() { cout << "A's destructor" << endl; }
};
class B
{ public:
    B() { cout << "B's default constructor" << endl; }
    B(const B&) { cout << "B's copy constructor" << endl; }
    ~B() { cout << "B's destructor" << endl; }
};
class C: public B
{    A a;
    public:
        C() { cout << "C's default constructor" << endl; }
        C(const C&) { cout << "C's copy constructor" << endl; }
        ~C() { cout << "C's destructor" << endl; }
};
void func1(C x) { cout << "func1" << endl; }
void func2(C &x) { cout << "func2" << endl; }
int main() {
    C c;           // Section 1
    func1(c);      // Section 2
    func2(c);      // Section 3
    return 0;      // Section 4
}
输出结果为：
// Section 1      // Section 2      // Section 3      // Section 4
```

23. 请写出下面程序的运行结果：（本题 4 分）

```
class A
{ public:
    A() { cout << "A's default constructor" << endl; f(); }
    A(const A&) { cout << "A's copy constructor" << endl; f(); }
    ~A() { cout << "A's default destructor" << endl; }
    virtual void f() { cout << "A's f" << endl; }
    void g() { cout << "A's g" << endl; }
    void h() { f(); g(); }
};

class B: public A
{ public:
    B() { cout << "B's default constructor" << endl; }
    B(const B&) { cout << "B's copy constructor" << endl; }
    ~B() { cout << "B's default destructor" << endl; }
    void f() { cout << "B's f" << endl; }
    void g() { cout << "B's g" << endl; }
};

void func1(A x) { x.f(); x.g(); x.h(); }
void func2(A &x) { x.f(); x.g(); x.h(); }

int main() {
    A a;                // Section 1
    A *p = new B;        // Section 1
    func1(a);            // Section 2
    func2(*p);           // Section 3
    delete p;            // Section 4
    return 0;            // Section 4
}

输出结果为:
// Section 1           // Section 2           // Section 3           // Section 4
```

### 三. 简答题

- 24. 指针类型主要用于什么场合？引用类型与指针类型相比，其优势在哪里？
- 25. 对类成员的访问，C++提供了哪些访问控制？
- 26. 拷贝构造函数的作用是什么？何时会调用拷贝构造函数？
- 27. 在C++中，三种继承方式各有什么作用？

#### 四. 程序设计题

28. 编写程序，使用如下的抽象类 **Shape**，由它派生 3 个类：**Square**(正方形)、**Trapezoid**（梯形）和 **Triangle**（三角形），用虚函数分别计算这几种图形的面积。在 **main** 函数中，要求用基类指针的数组，使它每一个元素指向一个派生类对象，并最终求出派生类图形的面积之和。

```
class Shape
{ public:
    virtual double area() const=0;
};
```

```
void main()
{    Square se(5); Trapezoid td(2,5,4); Triangle te(5,8);
```

```
}
```