

C++期末考试模拟卷

考试时间：2 小时

一. 选择题(每小题 2 分, 共 30 分)

(1)关于 C++与 C 语言的关系的描述中错误的是()

- A. C 语言是 C++的一个子集
- B. C++的编译器可以编译 C 语言代码
- C. C++对 C 语言进行了一些改进
- D. C++和 C 语言都是面向对象的

(2)C++中, 函数原型不能标识()

- A. 函数的参数类型与数量
- B. 函数的默认参数
- C. 函数的处理流程
- D. 函数的返回值类型

(3)使用派生类的主要原因是()

- A. 提高代码的可重用性
- B. 提高程序的运行效率
- C. 加强类的封装性
- D. 实现数据的隐藏

(4)类模板在使用时实际是先将类模板实例化成一个具体的()

- A. 类
- B. 对象
- C. 函数
- D. 模板类

(5)该函数合适使用哪项技术进行效率优化()

```
int func(int a, int b, int c) { return 2 * (a + b) / c; }
```

- A. 默认参数
- B. 函数重载
- C. 函数内联
- D. 虚函数重写

(6)成员函数可声明为静态的, 条件是它的实现中不访问()

- A. 非 const 成员
- B. 非 virtual 成员
- C. 非 static 成员
- D. 非 public 成员

(7)程序在调用重载函数时, 往往根据一些条件确定哪个重载函数被调用, 这个过程名称以及不能作为依据的一项分别是()

- A. 静态联编; 返回值
- B. 实例化; 参数类型
- C. 静态联编; 名称
- D. 实例化; 参数数量



(8)下面哪个函数和 `int func(int a, int b)` 构成重载函数时一定不会产生二义性()

- A. `string func(int a, int b, int c = 10);`
- B. `int func(int a, int b = 2);`
- C. `string func(int& a, int &b);`
- D. `int func(string a, string b);`

(9)若 A、B 是两个类, $A \leftarrow B$ 且 A 和 B 各含有一个整型公有成员变量 n, 则()

- A. B 类中无法访问 A 类的 n
- B. B 类中可以通过 `A::n` 访问 A 类的 n
- C. A 类中可以通过 `B::n` 访问 B 类的 n
- D. B 类中必须通过 `B::n` 访问 B 类的 n

(10)一个类可包含析构函数的个数是()

- A. 0
- B. 1
- C. 无数个, 但要求名称相同且参数列表不同
- D. 无数个, 且名称和参数列表相同, 编译器根据情况决定调用哪个

(10)C++中进行文件操作时需要包含头文件()

- A. `<iostream>`
- B. `<stdlib>`
- C. `<file>`
- D. `<fstream>`

(11)对于语句 `std::cout << x << std::endl;` 描述错误的是()

- A. “<<” 称作流提取运算符
- B. “endl” 的作用是回车换行
- C. “x” 是一个变量
- D. “cout” 是一个对象

(12)在 C++中, 若 $A \leftarrow C$ 且 $B \leftarrow C$, 则下列说法正确的是()

- A. C++不允许这种情况
- B. C 对 A 采用的继承方式必须和对 B 采用的一样
- C. 若同时还有 $D \leftarrow A$ 且 $D \leftarrow B$, 则 C++不允许这种情况, 否则允许
- D. 若同时还有 $D \leftarrow A$ 且 $D \leftarrow B$, 则可以采用虚继承来避免二义性



(13)当需要打开 D 盘上的 xxk.dat 文件用于输入时, 定义文件流对象的语句为()

- A. fstream fin("D:\\xxk.dat" , 1)
- B. ifstream fin("D:\\xxk.dat" , ios::in)
- C. ofstream fin("D:\\xxk.dat")
- D. ifstream fin("A:\\xxk.dat" , ios::app)

(14)若要对类 AB 定义加号操作符重载成员函数, 实现两个 AB 类对象的加法, 并返回相加结果, 则该运算符函数的声明语句应为()

- A. AB operator+(AB& a, AB& b)
- B. AB operator+(AB& a)
- C. operator+(AB a)
- D. AB& operator+()

(15)给定如下两个类 BASE 和 DERIVED, 则 DERIVED 的构造函数合理的是()

```
class BASE {
private: int na;
public: BASE(int n) { na = n; }
};

class DERIVED : protected BASE {
private: int nb;
public: // 略...
};
```

- A. DERIVED(int n1, int n2) : A(n1), nb(n2) { ... }
- B. DERIVED(int n1, int n2) : n1(na), n2(nb) { ... }
- C. DERIVED(int n1, int n2) : nb(n2) { A(n1); }
- D. DERIVED(int n1, int n2) : A(n1) { nb(n2); }



二. 填空题(每空 1 分, 共 10 分)

- (1)定义类的动态对象数组需要使用_____运算符, 此操作只能自动调用该类的_____构造函数对其中的对象进行构造。
- (2)一个类的拷贝构造函数应接收的参数为_____。
- (3)若需要指明函数 void func(); 为类 AB 的友元函数, 则应在类 AB 的定义中加入一条语句: _____。
- (4)在 C++ 中, 将不同功能库的类与函数分模块进行实现并提供_____供其他模块包含使用; 为了避免来自不同库的同名符号产生二义性, 应使用_____加以区分。
- (5)异常处理目的是应对程序在_____(填编译或运行)时有可能遇到的意外错误情况, 避免程序直接中断, 给程序一个机会在_____语句块中对意外情况进行处理。
- (6)一个类由于层级太宽泛而无法确定其某些行为的具体执行方式, 需要留待更精确的子类来描述, 则应将这些行为声明为_____, 这样的类称为_____。

三. 判断题(每题 1 分, 共 10 分)

- (1)在基类中被声明为虚函数的成员函数必须在每个派生类中继续声明为虚函数, 才能具有多态的特征()
- (2)函数重载技术的加入使得 C++ 和 C 语言有了本质区别()
- (3)引用就是某个变量的别名, 对引用的操作, 实质上就是对被引用的变量的操作()
- (4)通过抽象分析得到的类的属性和行为可以用 UML 类图进行描述()
- (5)友元函数是类的成员函数, 所以可以存取或修改该类中的私有成员()
- (6)就算被 inline 关键字声明的函数也不一定会被内联()
- (7)同样存储整型数据 65535, 通过文本文件存储比通过二进制文件存储更节省空间()
- (8)块 catch(...){ } 应置于多个 catch 块的最前面()
- (9)调用函数 template <typename T> T func(T a); 时可以直接写 func(n); 让编译器自行推断 T 所指代的类型()
- (10)为了控制读写流程, 类应为私有数据成员提供读写函数, 对于数据成员 xxx, 其读写函数的名称必须叫做 get_xxx 和 set_xxx ()



四. 简答题(共 15 分)

(1)请简述 C++ 源代码如何在计算机上运行(3 分)

(2)请简述 overload 和 override 的异同(4 分)

(3)若程序员没有定义拷贝构造函数, 则编译器自动生成一个缺省的拷贝构造函数, 它可能会产生什么问题(4 分)

(4)请简述为何 C++ 除兼容 C 语言 malloc/free 之外仍要引入 new/delete(4 分)



五. 代码阅读题(每题 3 分, 共 12 分)

(1)下列代码段的输出结果是:

```
template <typename KK> KK func(KK a, KK b, KK c) {
    return a > b ? (a > c ? a : c) : (b > c ? b : c);
}
cout << func(1, 2, 3) << func('a', 'b', 'C');
```

(2)下列代码段的输出结果是:

```
int m = 1, n = 3;
n = ++m * n--;
cout << n;
```

(3)下列代码段的输出结果是:

```
class A {
public:
    string s;
    A(string s) : s(s) { }
    A operator* (int n) {
        string t;
        for (int i = 0; i < n; i++) { t += s; }
        return A(t);
    }
};
A a1("abc");
A a2 = a1 * 3;
cout << a2.s;
```



(4)若存在下列函数定义, 则调用 fun2(10); 的输出结果是:

```
void fun1(int a) {
    try { if(a % 3 != 0) throw(a); }
    catch (string e) { cout << e << endl; }
    cout << "continued in fun1";
}

void fun2(int a) {
    try { fun1(a); }
    catch (int e) { cout << e << endl; }
    cout << "continued in fun2";
}
```

六. 程序补全题(共 8 分)

(1)(2 分)请合理地补全类 A 的构造函数以及数据成员 n 的 setter 函数

```
class A {
private:
    int n;
public:
    A(int n) {
        _____;
    }
    int get_n() const {
        return n;
    }
    void set_n(const int n) {
        _____;
    }
};
```



(2)(3 分)请补全下列代码，避免可能的资源泄露

```
class A {
public:
    A() {}
    _____(补充 A 的析构函数)
};

class B : protected A {
    int* ptr;
    ofstream fout;
public:
    B() {
        ptr = new int[100];
        fout.open("myFile.txt", ios::out);
    }
    ~B() {
        delete ptr;
        _____;
    }
    // B 的其他成员函数 ...
};

A* p = (A*) new B();
// 使用 p ...
_____;
```



(3)(3 分)请补全下列代码，使函数 func 可以让各种动物都进行自我介绍

```
class Animal {
public: virtual void say() = 0;
};

class Cat : public Animal{
public: void say() { cout << "I'm a cat" << endl; }
};

class Dog : public Animal {
public: virtual void say() { cout << "I'm a dog" << endl; }
};

class Corgi : public Dog {
public: void say() { cout << "I'm a corgi dog" << endl; }
};

void func(_____) {
    p->say();
}

int main() {
    Corgi corgi1, corgi2;
    Cat cat1, cat2, cat3;
    Dog dog1, dog2;
    func(_____); // 让 cat1 进行自我介绍
    func(_____); // 让 corgi2 进行自我介绍
    // 让其他动物进行自我介绍...
}
```



七. 程序设计题(共 15 分)

(1)(7 分)请实现一个类 A, 要求通过一个静态成员变量记录其实时对象数量, 并编写合理的测试代码来动态创建和销毁 A 类对象以及检查对象计数。

(2)(8 分)请实现一个图形类, 其包含属性颜色以及求解图形周长和面积的方法, 并实现其两个子类圆形和矩形, 其中圆形有属性半径, 矩形有属性长和宽。请注意根据事物特征按需使用虚函数等技术。请注意需要分别在 header.h 和 implement.cpp 中编写上述类的声明和实现, 并在 main.cpp 中提供合理的测试代码。

header.h:

implement.cpp:

main.cpp:

