



厦门大学《C++程序设计》课程复习试卷

● 请将答案按序写在学校统一印制的专用答题卷上，写在本卷或自备纸上者一律不得分。

一. 单选题 (含 15 个小题, 每小题 2 分, 计 30 分)

1. 以下哪个选项不全是 C++ 关键字的 (D)
A. namespace, float, double
B. private, inline, switch
C. class, union, enum
D. protect, virtual, static
2. 下列关于引用的叙述正确的是 (A)
☒ A. 作为函数参数时, 引用类型的实参是一个变量的别名, 而指针类型的实参是一个变量的地址。
☐ B. 变量引用在实参时使用 &a 的形式
☐ C. 引用只能用常量初始化
☐ D. 除了在定义时指定的被引用变量外, 引用类型还可以引用其他变量。
B
3. 下列关于 new 运算符的描述中, 哪个是错误的 (C)
☒ A. 它可以用来动态创建对象和对象数组
☐ B. 使用 new 创建的 int 型数组 p[10], 可以用 "delete p;" 来释放空间
☒ C. 使用它创建对象时要调用构造函数
☐ D. new 创建的动态变量的空间是在堆区中分配的
delete[] p;
4. 下面代码编译时不会报错的语句是 (B)
const int *p; 常量指针变量
int *q;
const int x=0;
int y;
☒ (1) p = &x; *p = 1
☒ (2) q = &y; *q = 1
(3) q = &x;
(4) p = &y;
(5) p = &x; q=p;
A. (1)(2) B. (2)(3) C. (2)(4) D. (2)(5)

5. 下列有关运算符重载的叙述中, 正确的是 (D)
- A. 重载运算符时可以改变其结合性
 - B. C++中可以通过运算符创造新的运算符
 - C. C++中所有的运算符都可以作为非成员函数重载
 - D. 运算符重载是多态的一种表现

6. 对于下面的类 A、B、C 和 D

```
class A{
    int x;
public:
    A(int i){x=i;}
};
class B: virtual public A{
    int y;
public:
    B(int i): A(1) {y=i;}
};
class C: virtual public A{
    int z;
public:
    C(int i): A(2) {z=i;}
};
class D: public B, public C {
    int m;
public:
    D(int i, int j, int k): C(j), B(i), A(3) {m=k;}
};
.....
D d(1,2,3);
```

创建 D 类对象 d 时, 所调用的构造函数及它们的执行顺序是 (C)

- A. D ()、B ()、C ()、A ()
 - B. D ()、C ()、B ()、A ()
 - C. A ()、B ()、C ()、D ()
 - D. A ()、C ()、B ()、D ()
7. 如果 myclass 类定义了拷贝构造函数和一个整型参数的构造函数, 还重载了赋值运算符。那么语句 myclass obj=10, 会 (C)
- A. 调用拷贝构造函数
 - B. 调用赋值运算符
 - C. 调用整型参数的构造函数
 - D. 引起编译错误

8. 关于友元, 下列说法错误的是 (A)
- A. 如果类 A 是类 B 的友元, 那么类 B 也是类 A 的友元
 - B. 如果函数 fun() 被说明为类 A 的友元, 那么在 fun() 中可以访问类 A 的私有成员
 - C. 友元关系不能被继承
 - D. 友元是数据保护和数据访问效率之间的一种折衷方案

9. 关于模板, 下列表述不正确的是 (B)
- A. 函数模板是指带有类型参数的函数定义
 - B. 函数模板只能带有类型参数
 - C. 类模板的实例化需要在程序中显式地指出
 - D. 类模板中的静态成员仅属于实例化后的类 (模板类)

10. 假定 AB 为一个类, 则执行 “AB a(2), b[3], *p[4];” 语句时调用该类构造函数的次数为 (B)

A. 3
B. 4
C. 5
D. 9

11. 有如下函数模板定义

```
template<class T>
T func( T x, T y)
{ return x*x+y*y; }
```

在下列 func 的调用中不正确的是

A. func(3, 5);
B. func(3, 5.5);
C. func<double>(3, 5);
D. func<int>(3, 5.5);

12. 下列有关输入/输出 (I/O) 的说法中不正确的是 (D)
- A. 在 C++ 中, 输入/输出不是语言定义的成分
 - B. 在 C++ 中, 输入/输出操作是一种基于字节流的操作
 - C. 对自定义的类重载插入操作符 “<<” 和抽取操作符 “>>” 时, 须作为全局函数来重载
 - D. 文件输入操作是指把计算机内存中的数据写入到外存中的文件里

13. 派生类的构造函数的成员初始化列表中, 不能包含 (C)
- A. 基类的构造函数
 - B. 基类的对象初始化
 - C. 派生类对象的初始化
 - D. 派生类中一般数据成员的初始化

14. 有如下类定义

```
class Base{
    int k;
    public:
    void set(int n){k=n;}
    int get() const {return k;}
};
class Derived: protected Base{
    protected:
    int j;
    public:
    void set(int m, int n){Base::set(m); j=n;}
    int get() const {return Base::get()+j;}
};
```

则类 Derived 中保护成员个数是 (B)

- A. 4
- B. 3
- C. 2
- D. 1

15. 下面语句 (1) 和 (2) 中哪句有错? A

```
#include<iostream>
class C{
public:
    void func(const C &c) const{
        std::cout<<c.d; // (1)
    }
private:
    int d;
};
int main(){
    C c;
    std::cout<<c.d; // (2)
}
```

- A. (1)正确, (2)错误
- B. (1)错误, (2) 正确
- C. 都正确
- D. 都错误

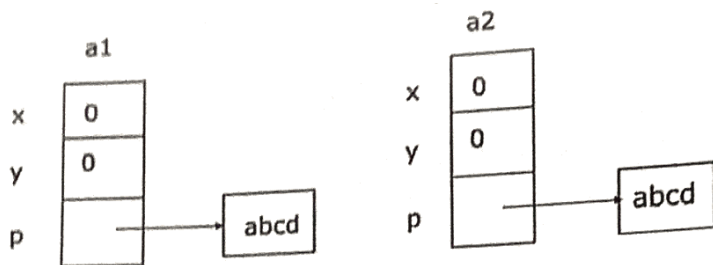
二. 程序分析题(含 5 个小题, 计 20 分)

16. 填写程序, 完成利用引用类型实现两个数的交换。(本题 4 分)

```
#include <iostream>
using namespace std; int &t1, int &t2
void swap(____(1)____) //交换两个 int 型变量的值
{
    int t;
    ____ (2) ____ t = t1;
    t1 = t2;
    t2 = t;
}
int main()
{
    int a=0, b=1;
    cout << a << ',' << b << endl; //结果为: 0,1
    swap(a,b);
    cout << a << ',' << b << endl; //结果为: 1,0
    return 0;
}
```

17. 填写程序, 以得到如图所示的结果。(本题 4 分)

```
class A
{
    int x,y;
    char *p;
public:
    A(char *str)
    {
        x = 0; y = 0;
        p = new char[strlen(str)+1];
        strcpy(p,str);
    }
    ~A() { delete [] p; p=NULL; }
};
A::A(____(1)____) const A &a
{
    x = a.x;
    y = a.y;
    ____ (2) ____ p = new [strlen(a.p)+1];
    strcpy(p,a.p);
}
.....
A a1("abcd");
A a2(a1);
```

18. 假设 B 是 A 的 public 继承的派生类, f 是 A 类的 public 成员函数, g 是 B 类新定义的 public 成员函数。阅读以下代码并回答问题。(本题 4 分)

```
A a;
B b;
a.g(); // (1)
A *p = &b; // (2)
b = a; // (3)
void func1(A *p);
func1(&b); // (4)
```

请回答问题:

1. 语句(1)是否合法? *不合法*
2. 语句(2)是否合法? *合法*
3. 语句(3)是否合法? *不合法*
4. 语句(4)是否合法? *不合法*

19. 阅读下列程序, 写出程序具体调用函数。(本题 4 分)

```
#include <iostream>
using namespace std;
```

```
class A{
public:
    A() { f(); }
    virtual ~A();
    virtual void f();
    void g();
    void h() { f(); g(); }
};
```

```
class B: public A{
public:
    ~B();
    void f();
    void g();
};
```

```
void main(){
```

```
    B b;
```

```
    A *p;
```

```
    p=&b;
```

```
    p->f();    //调用____(1)____
```

```
    p->A::f(); //调用____(2)____
```

```
    p->g();    //调用____(3)____
```

```
    p->h();    //调用____(4)____
```

```
    p = new B; //调用 B 的默认构造函数
```

```
    ...
```

```
    delete p; //调用~B()和~A()
```

```
}
```

Handwritten notes and corrections:

- Original: `p->f();` //调用____(1)____
- Correction: `p->f();` //调用____(1)____ (with `B::f` written above)
- Original: `p->A::f();` //调用____(2)____
- Correction: `A::f` (written above)
- Original: `p->g();` //调用____(3)____
- Correction: `B::g` (written above)
- Original: `p->h();` //调用____(4)____
- Correction: `A::h` (written above)

20. 完成如下程序。(本题 4 分)

```
#include <iostream>
```

```
using namespace std;
```

```
template <__class T>
```

```
class Stack
```

```
{    T buffer[size];
```

```
    int top;
```

```
public:
```

```
    Stack() { top = -1; }
```

```
    bool push(const T &x)
```

```
    {
```

```
        if (top == __size-1)
```

```
        {    cout << "Stack is overflow.\n";
```

```
            return false;
```

```
        }
```

```
        else
```

```
        {    top++; buffer[top] = x ;
```

```
            return true;
```

```
        }
```

```
    }
```

```
    bool pop(T &x)
```

```
    {
```

```
        if (top == -1)
```

```
        {    cout << "Stack is empty.\n";
```



```

        return false;
    }
    else { x=buffer[top]
        (3); top--;
        return true;
    }
}

};
int main()
{
    double x; double
    Stack< (4) > st1; // st1 为元素个数为 100 的 double 型栈
    st1.push(10.0);
    st1.pop(x);
    .....
}

```

三. 简答题 (含 4 个小题, 每小题 5 分, 计 20 分)

21. 与过程式程序设计相比, 面向对象程序设计的优势有哪些?

22. 什么是构造函数? 什么是拷贝构造函数? 它们有何区别?
构造函数是定义对象时, 对数据成员进行初始化的函数。
拷贝构造是函数。

23. C++ 怎样实现消息的动态绑定, 请简单说明下实现过程。
通过关键字 virtual 进行声明

24. C++ 标准模板库 (STL) 中包含哪几类模板? 它们的作用分别是什么?

四. 设计题 (含 4 个小题, 计 30 分)

26. 定义一个日期类 Date, 它能表示: 年、月、日, 并提供以下操作:

- (1) Date (int y, int m, int d); //构造函数
- (2) set(int y, int m, int d); //调整日期
- (3) display(); //显示日期。 (本题 7 分)

26. 把在链表中插入一个新结点的操作写成一个函数:

bool insert(Node *h, int a, int pos);

其中, h 为表头指针, a 为要插入的结点的值, pos (≥ 0) 表示插入位置。

```

class Date {
private:
    int year, month, day;
public:

```

```

    Date(int y, int m, int d)
    { year=y; month=m;
      day=d; }

```

```

    void set(int y, int m,
             int d)
    { year=y; month=m;
      day=d; }

```

```

    void display() {
        cout << y << "年" << month << "月"
        << day << "日" << endl; }
}

```



```

bool insert(Node* h, int a, int pos)
{
    Node* p = h;
    if(a == 0) { Node* q = new Node(a); q->data = a; q->next = p; return true; }
    while(--a) { p = p->next; if(p->next == null) return false; }
    Node* q = new Node(a); q->data = a; q->next = p->next; p->next = q; return true;
}

```

当 pos 为 0 时表示在表头插入；否则，表示在第 pos 个结点的后面插入。
 作成功返回 true，失败返回 false。（本题 7 分）

27. 编写程序，定义一个抽象类 Shape（如下），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。（本题 8 分）

```

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

```

28. 编写一个通讯录输入/输出程序。该程序从键盘输入通讯录，然后把它保存在文件中。通讯录的内容包括：序号、姓名、地址以及电话。要求在程序中把通讯录设计为一个类，通过重载操作符">>"和"<<"来实现通讯录的输入/输出。（本题 8 分）

```

27. #include <iostream>
using namespace std;
class Shape {
public:
    virtual double area() const = 0;
};
class Square: public Shape {
private:
    double x;
public:
    double area() const { return x*x; }
};
class Trapezoid: public Shape {

```