

**2006 年 4 月全国计算机等级考试二级 C++ 语言程序设计笔试试卷**

一、选择题((1)~(35)每小题 2 分, 共 70 分)

下列各题 A)、B)、C)、D)四个选项中, 只有一个选项是正确的, 请将正确选项涂写在答题卡相应位置上, 答在试卷上不得分。

(1)下列选项不属于结构化程序设计方法的是

- A)自顶向下 B)逐步求精  
C)模块化 D)可复用

(2)两个或两个以上模块之间关联的紧密程度称为

- A)耦合度 B)内聚度  
C)复杂度 D)数据传输特性

(3)下列叙述中正确的是

- A)软件测试应该由程序开发者来完成  
B)程序经调试后一般不需要再测试  
C)软件维护只包括对程序代码的维护  
D)以上三种说法都不对

(4)按照“后进先出”原则组织数据的数据结构是

- A)队列 B)栈  
C)双向链表 D)二叉树

(5)下列叙述中正确的是

- A)线性链表是线性表的链式存储结构  
B)栈与队列是非线性结构  
C)双向链表是非线性结构  
D)只有根结点的二叉树是线性结构

(6)对如下二叉树

进行后序遍历的结果为

- A)ABCDEF B)DBEAFC  
C)ABDECF D)DEBFCA

(7)在深度为 7 的满二叉树中, 叶子结点的个数为

- A)32 B)31 C)64 D)63

(8)“商品”与“顾客”两个实体集之间的联系一般是

- A)一对一 B)一对多  
C)多对一 D)多对多

(9)在 E-R 图中, 用来表示实体的图形是

- A)矩形 B)椭圆形 C)菱形 D)三角形

(10)数据库 DB、数据库系统 DBS、数据库管理系统 DBMS 之间的关系是

- A)DB 包含 DBS 和 DBMS B)DBMS 包含 DB 和 DBS  
C)DBS 包含 DB 和 DBMS D)没有任何关系

(11)下列关于 C++函数的说明中, 正确的是

- A)内联函数就是定义在另一个函数体内部的函数  
B)函数体的最后一条语句必须是 return 语句  
C)标准 C++要求在调用一个函数之前, 必须先声明其原型  
D)编译器会根据函数的返回值类型和参数表来区分函数的不同重载形式

(12)由于常对象不能被更新, 因此

- A)通过常对象只能调用它的常成员函数

B)通过常对象只能调用静态成员函数

C)常对象的成员都是常成员

D)通过常对象可以调用任何不改变对象值的成员函数

(13)下列关于虚函数的说明中，正确的是

A)从虚基类继承的函数都是虚函数 B)虚函数不得是静态成员函数

C)只能通过指针或引用调用虚函数 D)抽象类中的成员函数都是虚函数

(14)运算符重载是对已有的运算符赋予多重含义，因此

A)可以对基本类型(如 int 类型)的数据，重新定义“+”运算符的含义

B)可以改变一个已有运算符的优先级和操作数个数

C)只能重载 C++中已经有的运算符，不能定义新运算符

D)C++中已经有的所有运算符都可以重载

(15)模板对类型的参数化提供了很好的支持，因此

A)类模板的主要作用是生成抽象类

B)类模板实例化时，编译器将根据给出的模板实参生成一个类

C)在类模板中的数据成员都具有同样类型

D)类模板中的成员函数都没有返回值

(16)下列关于 C++流的说明中，正确的是

A)与键盘、屏幕、打印机和通信端口的交互都可以通过流类来实现

B)从流中获取数据的操作称为插入操作，向流中添加数据的操作称为提取操作

C)cin 是一个预定义的输入流类

D)输出流有一个名为 open 的成员函数，其作用是生成一个新的流对象

(17)下列符号中可以用作 C++标识符的是

A)\_radius B)foo~bar C)else D)3room

(18)下列语句中，错误的是

A)const int buffer=256; B)const int temp;

C)const double \*point; D)const double \*rt=new double(5.5);

(19)有如下程序：

```
int x=3;
do{
    x -= 2;
    cout<<
}while(!(--x));
```

执行这个程序的输出结果是

A) 1 B) 3 0 C) 1-2 D)死循环

(20)有如下程序：

```
#include
using namespace std;
int main()
{
    int *p;
    *p = 9;
    cout<<"The value at p:"<<*p;
    return 0;
}
```

编译运行程序将出现的情况是

- A)够编译时出现语法错误，不能生成可执行文件
- B)运行时一定输出：The value at p: 9
- C)运行时一定输出：The value at p: \*9
- D)运行时有可能出错

(21)有如下程序：

```
#include
using namespace std;
int main()
{
    void function(double val);
    double val;
    function(val);
    cout<
    return 0;
}
void function(double val)
{
    val = 3;
}
```

编译运行这个程序将出现的情况是

- A)编译出错，无法运行 B)输出：3
- C)输出：3.0 D)输出一个不确定的数

(22)有如下类定义：

```
class AA
{
    int a;
public:
    int getRef() const{return &a;} //①
    int getvalue() const{return a;} //②
    void set(int n) const{a=n;} //③
    friend void show(AA aa) const {cout< //④
};
```

其中的四个函数定义中正确的是

- A)① B)② C)③ D)④

(23)有如下程序：

```
#include +
using namespace std;
#include
using namespace std;
class Base
{
public:
    void fun(){cout<<"Base::fun"<
```

```
};  
class Derived:public Base  
{  
public:  
void fun()  
{  
cout<<"Derived::fun"<  
}  
};  
int main()  
{  
Derived d;  
d.fun();  
return 0;  
}
```

已知其执行后的输出结果为：

Base::fun

Derived::fun

则程序中下划线处应填入的语句是

A)Base.fun(); B)Base::fun() C)Base->fun() D)fun();

(24)下面是重载为非成员函数的运算符函数原型，其中错误的是

A)Fraction operator + (Fraction, Fraction);

B)Fraction operator - (Fraction);

C)Fraction& operator = (Fraction&, Fraction);

D)Fraction& operator += (Fraction&, Fraction);

(25)有如下函数模板：

template

T cast(U u){return u;}

其功能是将 U 类型数据转换为 T 类型数据。已知 i 为 int 型变量，下列对模板函数 cast 的调用中正确的是

A)cast(i); B)cast<>(i);

C)cast(i); D)cast (i);

(26)若磁盘上已存在某个文本文件，其全路径文件名为 d:\ncre\test.txt，下列语句中不能打开该文件的是

A)ifstream file("d:\ncre\test.txt");

B)ifstream file("d:\ncre\test.txt");

C)ifstream file; file.open("d:\ncre\test.txt");

D)ifstream\* pFile=new ifstream("d:\ncre\test.txt");

(27)有如下类定义：

class Foo

{

public:

Foo(int v):value(v){} //①

~Foo(){} //②

```
private:
```

```
Foo(){ } //③
```

```
int value = 0; //④
```

```
};
```

其中存在语法错误的行是

A)① B)② C)③ D)④

(28)有如下类定义：

```
class Point
```

```
{
```

```
int x_, y_;
```

```
public:
```

```
Point():x_(0), y_(0){ }
```

```
Point(int x, int y = 0):x_(x),y_(y){ }
```

```
};
```

若执行语句

```
Point a(2), b[3], *c[4];
```

则 Point 类的构造函数被调用的次数是

A)2 次 B)3 次 C)4 次 D)5 次

(29)有如下类定义：

```
class Test
```

```
{
```

```
public:
```

```
Test(){ a = 0; c = 0; } //①
```

```
int f(int a)const{ this->a = a; } //②
```

```
static int g(){ return a; } //③
```

```
void h(int b){ Test::b = b; }; //④
```

```
private:
```

```
int a;
```

```
static int b;
```

```
const int c;
```

```
};
```

```
int Test::b = 0;
```

在标注号码的行中，能被正确编译的是

A)① B)② C)③ D)④

(30)有如下程序：

```
#include
```

```
using namespace std;
```

```
class A
```

```
{
```

```
public:
```

```
A(){ cout<<"A"; }
```

```
~A(){ cout<<"~A"; }
```

```
};
```

```
class B:public A
```

```
{
A* p;
public:
B(){cout<<"B"; p = new A();}
~B(){cout<<"~B"; delete p;}
};
int main()
{
B obj;
return 0;
}
```

执行这个程序的输出结果是

A)BAA~A~B~A B)ABA~B~A~A

C)BAA~B~A~A D)ABA~A~B~A

(31)有如下程序：

```
#include
using namespace std;
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;
}
```

其中有语法错误的语句是

A)①②③④ B)①②③ C)②③④ D)①④

(32)有如下程序：

```
#include
```

```
using namespace std;
class Base
{
public:
void fun1(){cout<<"Base\n";}
virtual void fun2(){cout<<"Base\n";}
};
class Derived:public Base
{
public:
void fun1(){cout<<"Derived\n";}
void fun2(){cout<<"Derived\n";}
};
void f(Base &b){b.fun1();b.fun2();}&, It;, /P>
int main()
{
Derived obj;
f(obj);
return 0;
}
```

执行这个程序的输出结果是

A)Base B)Base C)Derived D)Derived

Base Derived Base Derived

(33)有如下程序：

```
#include
using namespace std;
class Complex
{
double re, im;
public:
Complex(double r, double i):re(r), im(i){}
double real() const{return re;}
double image() const{return im;}
Complex& operator +=(Complex a)
{
re += a.re;
im += a.im;
return *this;
};
ostream &operator<<(ostream& s,const Complex& z)
{
return s<<'\n';
}
```

```
int main()
{
    Complex x(1, -2), y(2, 3);
    cout<<(x += y)<
    return 0;
}
```

执行这个程序的输出结果是

A)(1, -2) B)(2, 3) C)(3, 5) D)(3, 1)

(34)有如下程序：

```
#include
#include
using namespace std;
int main()
{
    cout.fill('*');
    cout<
    return 0;
}
```

执行这个程序的输出结果是

A)123\*OK B)123\*OK\*\* C)\*123OK D)\*123\*\*OK

(35)有如下类声明：

```
class SAMPLE
{
    int n;
public:
    SAMPLE (int i = 0): n(i){}
    void setvalue(int n0);
};
```

下列关于 setvalue 成员函数的实现中，正确的是

- A)SAMPLE::setvalue(int n0){n = n0;}
- B)void SAMPLE::setvalue(int n0){n = n0;}
- C)void setvalue(int n0){n = n0;}
- D)setvalue(int n0){n = n0;}

二、填空题(每空 2 分，共 30 分)

请将每一个空的正确答案写在答题卡【1】～【15】序号的横线上，答在试卷上不得分。

(1)对长度为 10 的线性表进行冒泡排序，最坏情况下需要比较的次数为 【1】。

(2)在面向对象方法中，【2】描述的是具有相似属性与操作的一组对象。

(3)在关系模型中，把数据看成是二维表，每一个二维表称为一个【3】。

(4)程序测试分为静态分析和动态测试。其中【4】是指不执行程序，而只是对程序文本进行检查，通过阅读和讨论，分析和发现程序中的错误。

(5)数据独立性分为逻辑独立性与物理独立性。当数据的存储结构改变时，其逻辑结构可以不变，因此，基于逻辑结构的应用程序不必修改，称为【5】。

(6)有一种只在循环语句或 switch 语句中使用的流程控制语句，表示该语句的关键字是【6】。

(7)程序中有如下语句



```
for(int i = 0; i<5; i++)cout<<*(p + i)<<" ";
```

能够依次输出 int 型一维数组 DATA 的前 5 个元素。由此可知，变量 p 的定义及初始化语句是 **【7】**。

(8)若下面程序运行时输出结果为

1, A, 10.1

2, B, 3.5

请将程序补充完整。

```
#include
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
void test(int, char, double 【8】 );
```

```
test(1, 'A', 10.1);
```

```
test(2, 'B');
```

```
return 0;
```

```
}
```

```
void test(int a, char b, double c)
```

```
{
```

```
cout<
```

```
}
```

(9)有如下类声明：

```
class MyClass
```

```
{
```

```
int i;
```

```
private: int j;
```

```
protected: int k;
```

```
public: int m, n;
```

```
};
```

其中，私有成员的数量为 **【9】**。

(10)有如下程序：

```
#include
```

```
using namespace std;
```

```
class Con
```

```
{
```

```
char ID;
```

```
public:
```

```
Con():ID('A'){cout<<1;}
```

```
Con(char ID):ID(ID){cout<<2;}
```

```
Con(Con& c):ID(c.getID()){cout<<3;}
```

```
char getID() const{return ID;}
```

```
};
```

```
void show(Con c){cout<
```

```
int main()
```

```
{
```

```
Con c1;
```

```
show(c1);  
Con c2('B');  
show(c2);  
return 0;  
}
```

执行上面程序的输出是 **【10】**。

(11)有如下程序：

```
#include  
using namespace std;  
class ONE  
{  
public:  
virtual void f(){cout<<"1";}  
};  
class TWO:public ONE  
{  
public:  
TWO(){cout<<"2";}  
};  
class THREE:public TWO  
{  
public:  
virtual void f(){TWO::f(); cout<<"3";}  
};  
int main()  
{  
ONE aa, *p;  
TWO bb;  
THREE cc;  
p = &cc;  
p->f();  
return 0;  
}
```

执行上面程序的输出是 **【11】**。

(12)有如下程序：

```
#include  
using namespace std;  
class PARENT  
{  
public:  
PARENT(){cout<<"PARENT";}  
};  
class SON:public PARENT  
{  

```

```
public:
SON(){cout<<"SON";}
};
int main()
{
SON son;
PARENT *p;
p = &son;
return 0;
}
```

执行上面程序的输出是 【12】 。

(13)若以非成员函数形式，为类 Bounce 重载“!”运算符，其操作结果为一 bool 型数据，则该运算符重载函数的原型是 【13】 。

(14)下面是复数类 complex 的定义，其中重载的运算符“+”的功能是返回一个新的复数对象，其实部等于两个操作对象实部之和，其虚部等于两个操作对象虚部之和；请补充完整：

```
class complex
{
double real; //实部
double imag; //虚部
public:
complex(double r,double i):real(r),imag(i){ }
complex operator+(complex a)
{
return complex( 【14】 );
}
};
```

(15)已知一个函数模板定义为

```
template T1, typename T2>
T1 FUN(T2 n) {return n * 5.0;}
```

若要求以 int 型数据 7 为函数实参调用该模板函数，并返回一个 double 型数据，则该调用应表示为

【15】。