

全国计算机等级考试（NCRE）

二级 C++ 语言程序设计

样题及参考答案

➤ 样题

一、单项选择题

1、下列符号中可以用作 C++ 标识符的是

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

2、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);

3、有如下程序段：

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

运行时的输出结果是

- A) 1
- B) 3 0
- C) 1-2
- D) 死循环

4、有如下语句序列：

```
int k=1,num=26;
do {
    k*=num%10;
    num/=10;
} while(num);
```

运行后变量 k 的值是

- A) 2
- B) 12
- C) 60
- D) 18

5、if 语句的语法格式可以描述为：

格式 1: if(<条件>)<语句>

或

格式 2: if(<条件>)<语句 1>else<语句 2>

下列关于 if 语句语法格式的叙述中，错误的是

- A) <条件>部分可以是一个 if 语句，例如 if(if(a==0)...)...
- B) <语句>部分可以是一个 if 语句，例如 if(...if(...)...
- C) 如果在<条件>前加上逻辑非运算符并交换<语句 1>和<语句 2>的位置，语句功能不变
- D) <语句>部分可以是一个循环语句，例如 if(...while(...)...

6、下列关于 C++函数的叙述中，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) C++要求在调用一个函数之前，必须先对其进行原型声明或定义
- 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

7、运算符重载是对已有的运算符赋予多重含义，下列关于运算符重载的叙述中，正确的是

- 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

8、下列关于内联函数的叙述中，正确的是

- A) 内联函数在调用时发生控制转移
- B) 内联函数必须通过关键字 `inline` 来定义
- C) 内联函数是通过编译器来实现的
- D) 内联函数的函数体的最后一条语句必须是 `return` 语句

9、下列运算符中，不能重载为友元函数的是

- A) `= () [] ->`
- B) `+ - ++ --`
- C) `> < >= <=`
- D) `+= -= *= /=`

10、有如下程序：

```
#include <iostream>
using namespace std;
void function(double& val);
int main()
{
    double val=1;
    function(val);
    cout<<val;
    return 0;
}
void function(double& val)
{
    val=3;
    cout<<val;
}
```

运行时的输出结果是

- A) 11
- B) 13
- C) 31
- D) 33

11、有如下程序：

```
#include<iostream>
using namespace std;
void fun(int val) { cout<<"int"<<endl; }
void fun(double val) { cout<<"double"<<endl; }
void fun(int val1,double val2) { cout<<"int+double"<<endl; }
int main()
{
    fun(1);
    return 0;
}
```

下列叙述中，正确的是

- A) 运行时的输出结果是 int
- B) 运行时的输出结果是 double
- C) 运行时的输出结果是 int+double
- D) 程序编译出错，无法运行

12、有如下类定义：

```
class MyClass {
public:
    _____
private:
    int data;
};
```

若要为 MyClass 类重载流提取运算符>>，使得程序中可以用“cin>>obj;”的形式改变 MyClass 类的对象 obj 中数据成员 data 的值，则横线处缺失部分为

- A) friend ostream& operator>>(ostream& is,MyClass& a);
- B) friend ostream& operator>>(ostream& is,MyClass a);
- C) ostream& operator>>(ostream& is,MyClass& a);
- D) ostream& operator>>(ostream& is,MyClass a);

13、下列关于常对象的叙述中，正确的是

- A) 通过常对象只能调用常成员函数
- 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

14、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<<a;} //④  
};
```

在标注号码的行中，能被正确编译的是

- A) ①
- B) ②
- C) ③
- D) ④

15、下列关于 **this** 指针的叙述中，正确的是

- A) 类的成员函数都有 **this** 指针
- B) 类的友元函数都有 **this** 指针
- C) 任何与类相关的函数都有 **this** 指针
- D) 类的非静态成员函数都有 **this** 指针

16、下列关于虚函数的叙述中，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

17、下列关于基类和派生类关系的叙述中，正确的是

- A) 每个类最多只能有一个直接基类
- B) 派生类中的成员可以访问基类中的任何成员
- C) 基类的构造函数必须在派生类的构造函数体中调用
- D) 派生类除了继承基类的成员，还可以定义新的成员

18、有如下类定义：

```
class B{
public:
    void fun1(){}
private:
    void fun2(){}
protected:
    void fun3(){}
};
class D:public B{
protected:
    void fun4(){}
};
```

若 obj 是类 D 的对象，则下列语句中不违反访问控制权限的是

- A) obj.fun1();
- B) obj.fun2();
- C) obj.fun3();
- D) obj.fun4();

19、有如下类定义：

```
class Foo {
public:
    Foo(int v):value(v) {} //①
    ~Foo(int v) {} //②
private:
    Foo() {} //③
    int value; //④
};
```

其中存在语法错误的行是

- A) ①
- B) ②
- C) ③
- D) ④

20、有如下类定义：

```
class Point {  
    int xx,yy;  
public:  
    Point():xx(0),yy(0){}  
    Point(int x,int y=0):xx(x),yy(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

21、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

22、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 成员函数的定义中，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;}

23、有如下程序：

```
#include <iostream>
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;
}
```

运行时的输出结果是

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

24、有如下程序：

```
#include <iostream>
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

25、有如下程序：

```
#include <iostream>
using namespace std;
class Base1 {
public:
    Base1(int d){cout<<d;}
    ~Base1(){}
}
```

```

};
class Base2{
public:
    Base2(int d){cout<<d;}
    ~Base2(){}
};
class Derived:public Base1,Base2{
public:
    Derived(int a,int b,int c,int d):Base1(b),Base2(a),b1(d),b2(c){}
private:
    int b1;
    int b2;
};
int main(){
    Derived d(1,2,3,4);
    return 0;
}

```

运行时的输出结果是

- A) 1234
- B) 2134
- C) 12
- D) 21

26、有如下程序：

```

#include <iostream>
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();}
int main(){
    Derived obj;
    f(obj);
    return 0;
}

```

运行时的输出结果是

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

27、模板对类型的参数化提供了很好的支持，下列叙述中，正确的是

- 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) 类模板中的数据成员都具有同样的类型
- D) 类模板中的成员函数都没有返回值

28、有如下函数模板：

```

template<typename T,typename U>
T cast(U u){return u;}

```

其功能是将 U 类型数据转换为 T 类型数据。若已知 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<char*,int>(i);
- D) cast<double,int>(i);

29、若要建立文件流并打开当前目录下的文件 file.dat 用于输入，则下列语句中错误的是

- A) ifstream fin=ifstream.open("file.dat");
- B) ifstream *fin=new ifstream("file.dat");
- C) ifstream fin;fin.open("file.dat");
- D) ifstream *fn=new ifstream();fin->open("file.dat");

- 30、当使用 `ifstream` 流类定义一个流对象并打开一个磁盘文件时，文件的默认打开方式是
- A) `ios_base::in`
 - B) `ios_base::in|ios_base::out`
 - C) `ios_base::out`
 - D) `ios_base::in&ios_base::out`

二、基本操作题

31.打开考生文件夹下的解决方案文件 `proj1`，其中的主程序文件 `main.cpp` 中包含类 `MyClass` 的定义。程序中位于每个 `//ERROR*****found*****` 下的语句行有错误，请加以更正。更正后程序的输出应该是：

The value is 5

The value is 10

注意：只能修改每个 `//ERROR*****found*****` 下的那一行，不要改动程序中的其他内容。

注：操作题仅提供题干内容，素材文件略。

三、简单应用题

32、打开考生文件夹下的解决方案文件 `proj2`，其中的主程序文件 `main.cpp` 中定义了 `Employee` 类和 `Manager` 类。`Employee` 用于表示某公司的雇员，其属性包括姓名(`name`) 和工作部门(`dept`)。`Manager` 是 `Employee` 的公有派生类，用于表示雇员中的经理。除了姓名和工作部门之外，`Manager` 的属性还包括级别(`level`)。`Employee` 类的成员函数 `print` 负责输出雇员的信息；`Manager` 类的成员函数 `print` 负责输出经理的信息。请在程序中的横线处填写适当的代码，然后删除横线，以实现上述类定义。此程序的正确输出结果应为：

Name:Sally Smith

Dept:Sales

Level:2

注意：只能在横线处填写适当的代码，不要改动程序中的其他内容，也不能删除或移动 `"//*****found*****"`。

注：操作题仅提供题干内容，素材文件略。

四、综合应用题

33、打开考生文件夹下的解决方案文件 `proj3`，其中包含主程序文件 `main.cpp` 和用户定义的头文件 `Matrix.h`。程序中定义了一个用于表示矩阵的类 `Matrix`，`operator+`的功能是实现两个矩阵的加法运算。例如，若有两个矩阵

$$m1 = \begin{bmatrix} 9 & 40 & 29 & \dots \\ 5 & 0 & 0 & \dots \\ 47 & 46 & 26 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \quad m2 = \begin{bmatrix} 47 & 46 & 26 & \dots \\ 13 & 28 & 34 & \dots \\ 44 & 14 & 11 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

则 `m1` 与 `m2` 相加的和为

$$sum = \begin{bmatrix} 56 & 86 & 55 & \dots \\ 18 & 28 & 34 & \dots \\ 91 & 60 & 37 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

请编写 `operator+` 函数。

要求：

补充编制的内容写在 `*****333*****` 与 `*****666*****` 两行之间。不得修改程序的其他部分。

注意：

程序最后已经将结果输出到文件 `out.dat` 中。输出函数 `writeToFile` 已经编译为 `obj` 文件，并且在本程序中调用。

注：操作题仅提供题干内容，素材文件略。

➤ 参考答案

一、单项选择题

1.A	2.B	3.C	4.B	5.A
6.C	7.C	8.C	9.A	10.D
11.A	12.A	13.A	14.B	15.D
16.B	17.D	18.A	19.B	20.C
21.D	22.B	23.B	24.B	25.D
26.B	27.B	28.D	29.A	30.A

二、基本操作题

(略)

三、简单应用题

(略)

四、综合应用题

(略)