

# C++程序设计试题及答案解析（一）

## C++程序设计模拟试卷(一)

### 一、单项选择题

1. 编写C++程序一般需经过的几个步骤依次是（）

- A. 编辑、调试、编译、连接
- B. 编辑、编译、连接、运行
- C. 编译、调试、编辑、连接
- D. 编译、编辑、连接、运行

答案：B

解析：(P21)经过编辑、编译、连接和运行四个步骤。编辑是将C++源程序输入计算机的过程，保存文件名为cpp。编译是使用系统提供的编译器将源程序cpp生成机器语言的过程，目标文件为obj，由于没有得到系统分配的绝对地址，还不能直接运行。连接是将目标文件obj转换为可执行程序的过程，结果为exe。运行是执行exe，在屏幕上显示结果的过程。

2. 决定C++语言中函数的返回值类型的是（）

- A. return语句中的表达式类型
- B. 调用该函数时系统随机产生的类型
- C. 调用该函数时的主调用函数类型
- D. 在定义该函数时所指定的数据类型

答案：D

解析：(P51)函数的返回值类型由定义函数时的指定的数据类型决定的。A项的表达式值要转换成函数的定义时的返回类型。

3. 下面叙述不正确的是（）

- A. 派生类一般都用公有派生
- B. 对基类成员的访问必须是无二义性的
- C. 赋值兼容规则也适用于多重继承的组合
- D. 基类的公有成员在派生类中仍然是公有的

答案：D

解析：(P136)继承方式有三种：公有、私有和保护。多继承中，多个基类具有同名成员，在它们的子类中访问这些成员，就产生了二义性，但进行访问时，不能存在二义性。赋值兼容规则是指派生类对象可以当作基类对象使用，只要存在继承关系，所以单继承或多继承都适用。基类中的公有成员采用私有继承时，在派生类中变成了私有成员，所以D项错误。

4. 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，这实体也就是（）

- A. 类

- B.* 对象
- C.* 函数体
- D.* 数据块

答案: A

解析: (P39)类即数据和操作的组合体, 数据是类的静态特征, 操作是类具有的动作。

5. 在公有派生类的成员函数不能直接访问基类中继承来的某个成员, 则该成员一定是基类中的 ( )

- A.* 私有成员
- B.* 公有成员
- C.* 保护成员
- D.* 保护成员或私有成员

答案: A

解析: (P133)在派生类中基类的保护或者基类公有都可以直接访问, 基类的私有成员只能是基类的成员函数来访问。所以选择A项。

6. 对基类和派生类的关系描述中, 错误的是 ( )

- A.* 派生类是基类的具体化
- B.* 基类继承了派生类的属性
- C.* 派生类是基类定义的延续
- D.* 派生类是基类的特殊化

答案: B

解析: (P129)派生类的成员一个是来自基类, 一个来自本身, 所以派生类是基类的扩展, 也是基类的具体化和特殊化, 派生类是对基类扩展。B项基类不能继承派生类成员, 所以错误。

7. 关于this指针使用说法正确的是 ( )

- A.* 保证每个对象拥有自己的数据成员, 但共享处理这些数据的代码
- B.* 保证基类私有成员在子类中可以被访问。
- C.* 保证基类保护成员在子类中可以被访问。
- D.* 保证基类公有成员在子类中可以被访问。

答案: A

解析: (P86)this指针是隐藏的, 可以使用该指针来访问调用对象中的数据。基类的成员在派生类中能否访问, 与继承方式有关, 与this没有关系。所以选择A项。

8. 所谓多态性是指 ( )

- A.* 不同的对象调用不同名称的函数
- B.* 不同的对象调用相同名称的函数
- C.* 一个对象调用不同名称的函数

D. 一个对象调用不同名称的对象

答案: B

解析: (P167)多态性有两种静态多态性和动态多态性, 静态多态性是指调用同名函数, 由于参数的不同调用不同的同名函数; 动态多态性是指不同对象调用同名函数时, 由于对象不同调用不同的同名函数。多态性肯定具有相同的函数名, 所以选择B项。

9. 一个函数功能不太复杂, 但要求被频繁调用, 则应把它定义为 ()

A. 内联函数

B. 重载函数

C. 递归函数

D. 嵌套函数

答案: A

解析: (P59)内联函数特征代码少, 频繁调用, 执行效率高。重载函数解决统一接口的问题; 递归是子程序调用, 程序调用要耗费很多空间和时间, 循环/迭代都比递归有效率得多, 递归只是从形式上, 逻辑比较简洁。嵌套函数即反复调用, 速度较慢。所以选择A项。

10. 下面函数模板定义中不正确的是 ()

A. A

B. B

C. C

D. D

答案: A

解析: (P147)A项中F是一个返回Q类型的值, 而return中用返回类型作为返回值错误。所以选择A项。

11. 假设Class Y:public X, 即类Y是类X的派生类, 则说明一个Y类的对象时和删除Y类对象时, 调用构造函数和析构函数的次序分别为 ()

A. X,Y; Y,X

B. X,Y; X,Y

C. Y,X; X,Y

D. Y,X; Y,X

答案: A

解析: (P130)派生类构造函数必须对这三类成员进行初始化, 其执行顺序: 调用基类构造函数; 调用子对象的构造函数; 派生类的构造函数体。析构函数在执行过程中也要对基类和成员对象进行操作, 但它的执行过程与构造函数正好相反, 即对派生类新增普通成员进行清理; 调用成员对象析构函数, 对派生类新增的成员对象进行清理; 调用基类析构函数, 对基类进行清理, 所以选择A项。

12. 适宜采用inline定义函数情况是 ()

- A. 函数体含有循环语句
- B. 函数体含有递归语句
- C. 函数代码少、频繁调用
- D. 函数代码多、不常调用

答案: C

解析: (P59)内联函数具有程序代码少、频繁调用和执行效率高的特征, 所以选择C项。

13. 假定一个类的构造函数为A(int aa,int bb) {a=aa--;b=a\*bb;},则执行A x(4,5); 语句后, x.a和x.b的值分别为 ()

- A. 3和15
- B. 5和4
- C. 4和20
- D. 20和5

答案: C

解析: (P75)a=4,因为后减, b的值与a、bb相关, b=4\*5=20, 而与aa没有任何关系。

14. 在类中说明的成员可以使用关键字的是 ()

- A. public
- B. extern
- C. cpu
- D. register

答案: A

解析: extern用于声明外部变量的。register声明寄存器类型变量。无cpu类型。它们都不能声明类成员。public声明为公有访问权限, 所以选择A项。

15. 下列不能作为类的成员的是 ()

- A. 自身类对象的指针
- B. 自身类对象
- C. 自身类对象的引用
- D. 另一个类的对象

答案: B

解析: 类的定义, 如果有自身类对象, 使得循环定义, B项错误。在类中具有自身类的指针, 可以实现链表的操作, 当然也可以使用对象的引用。类中可以有另一个类的对象, 即成员对象。所以选择B选项。

16. 使用地址作为实参传给形参, 下列说法正确的是 ()

- A. 实参是形参的备份

B. 实参与形参无联系

C. 形参是实参的备份

D. 实参与形参是同一对象

答案: D

解析: (P51)地址作为实参, 表示实参与形参代表同一个对象。如果实参是数值, 形参也是普通变量, 此时形参是实参的备份。所以选择D项。

17. 下列程序的输出结果是 ( )

```
#include <iostream.h>
```

```
void main()
```

```
{int n [ ] [3] ={10,20,30,40,50,60};
```

```
int (*p) [3] ;
```

```
p=n;
```

```
cout<<p [0] [0] <<" , "<<*(p [0] +1)<<" , "<<(*p) [2] <<endl;}
```

A. 10, 30, 50

B. 10, 20, 30

C. 20, 40, 60

D. 10, 30, 60

答案: B

解析: 如果数组元素都是相同类型的指针, 则称这个数组为指针数组。指针数组一般用于处理二维数组。声明的格式为: <数据类型><(\*变量名)>< [元素个数] >。

p表示指向数组n的行指针。如果将指针的初始化(\*p) [3] =b;地址的等价形式:

p+i p [i] \*(p+i)都表示b数组第i+1行的第1个元素的首地址。

\*(p+i)+j p [i] +j &p [i] [j] 都表示b数组第i+1行、第j+1列元素的地址。

值的等价形式:

\*(\*(p+i)+j) \*(p [i] +j) p [i] [j] 都表示b数组第i+1、第j+1列元素的值。

所以题目分别访问p [0] [0] , p [0] [1] , p [0] [2] 。

18. 在C++中, 使用流进行输入输出, 其中用于屏幕输入 ( )

A. cin

B. cerr

C. cout

D. clog

答案: A

解析: (P193)(1)标准输入流cin: istream类的对象。(2)标准输出流cout: ostream类的对象。

(3)非缓冲型标准出错流cerr: ostream类的对象。(4)缓冲型标准出错流clog: ostream类的对象

19. 假定AA为一个类，a()为该类公有的函数成员，x为该类的一个对象，则访问x对象中函数成员a()的格式为（）

- A. x.a
- B. x.a()
- C. x->a
- D. (\*x) .a()

答案: B

解析: (P41)对象访问成员的方式为: 对象名.成员。指针可以有两种: (\*对象指针).成员或者对象指针->成员。A选项是访问数据成员，B项是访问成员函数。

20. 关于对象概念的描述中，说法错误的是（）

- A. 对象就是C语言中的结构变量
- B. 对象代表着正在创建的系统中的实体
- C. 对象是类的一个变量
- D. 对象之间的信息传递是通过消息进行的

答案: A

解析: (P37)A对象在C++中才有，包括数据和操作两项，而C中的变量只有数据，没有操作。所以A项错误。

二、填空题(本大题共20小题，每小题1分，共20分)请在每小题的空格中填上正确答案。

。错填、不填均无分。

1. C++的流库预定义了4个流，它们是cin、cout、clog和\_\_\_。

答案: (P193)cerr

【解析】 cin、cout、clog和cerr分别用于标准输入、输出、标准错误流（缓冲）和标准错误流（非缓冲）。

2. 每个对象都是所属类的一个\_\_\_。

答案: (P69)实例

【解析】 类是对象的抽象，对象是类的一个实例。

3. 在已经定义了整型指针ip后，为了得到一个包括10个整数的数组并由ip所指向，应使用语句\_\_\_。

答案: (P78)int \*ip=new int [10] ;

【解析】 new用来动态开辟空间。常用来产生动态数组及对象构造函数。

4. 函数模板中紧随template之后尖括号内的类型参数都要冠以保留字\_\_\_。

答案: (P145)class

【解析】 类模板的使用。template <class T>,也可以引入多参数的如: template <class T1, class T2,..., class Tn>

5. 定义类的动态对象数组时，系统只能够自动调用该类的\_\_\_构造函数对其进行初始化。

答案: (P80)无参

【解析】使用new创建对象数组，调用无参构造函数。

6. 表达式cout<<endl 还可表示为\_\_\_。

答案: ‘\n’

【解析】endl与字符常量 ‘\n’ 等价。

7. 在C++中，访问一个指针所指向的对象的成员所用的指向运算符是\_\_\_。

答案: ->

【解析】指针使用成员有两种方法：“->”指向运算符和“.”成员访问运算符。

8. 假如一个类的名称为MyClass，使用这个类的一个对象初始化该类的另一个对象时，可以用\_\_\_构造函数来完成此功能。

答案: (P80)复制或拷贝

复制或拷贝构造函数就是用对象初始化新的对象。

9. 对赋值运算符进行重载时，应声明为\_\_\_函数。

答案: (P183)类成员

【解析】运算符重载的方法有友元或者成员函数两种途径，但是赋值运算符只能使用成员函数的方法来实现。

10. 如果要把A类成员函数f () 且返回值为void声明为类B的友元函数，则应在类B的定义中加入的语句\_\_\_。

答案: (P109)friend void A::f();

【解析】成员函数作为另一个类的友元函数，格式为：friend 返回类型 类名::函数(形参)。

11. 下列程序段的输出结果是\_\_\_。

```
for(i=0,j=10,k=0;i<=j;i++j-=3,k=i+j);cout<<k;
```

答案: 4

【解析】for循环结构,三个表达式的作用，初始化、循环判断条件和循环变量变化。循环执行了三次，k的作用是计算i、j的和。

12. String 类的\_\_\_方法返回查找到的字符串在主串的位置。

答案: (P40)find

【解析】string类对象方法的find，查不到字符串，则返回-1。

13. int n=0;

```
while (n=1) n++;
```

while循环执行次数是\_\_\_。

答案: 无限次

【解析】=是赋值运算符，不是关系运算符，且不等0，所以死循环。

14. 控制格式输入输出的操作中，函数\_\_\_是用来设置填充字符。要求给出函数名和参数类型

答案: (P195)setfill(char)

〔解析〕 格式控制方法的使用，如setw，setfill等等。

15. C++语言支持的两种多态性分别是编译时的多态性和\_\_\_的多态性。

答案: (P167)运行时

〔解析〕 多态性包括静态的（编译时）多态性和动态的（运行时）多态性。

16. 设函数sum是由函数模板实现的，并且sum(3,6)和sum(4.6,8)都是正确的函数调用，则函数模板具有\_\_\_个类型参数。

答案: (P61)2

17. 执行下列代码

```
string str("HelloC++");
```

```
cout<<str.substr(5, 3);
```

程序的输出结果是\_\_\_。

答案: (P42)C++

〔解析〕 substr取子字符串，第1个参数表示要截取子串在字符串中的位置，第2个表示取多少个字符。

18. 在面向对象的程序设计中，将一组对象的共同特性抽象出来形成\_\_\_。

答案: (P38)类

〔解析〕 类是相似特征的对象抽象，对象是类的一个实例。

19. 定义类动态对象数组时，元素只能靠自动调用该类的\_\_\_来进行初始化。

答案: (P77)无参构造函数

〔解析〕 使用new 创建动态对象数组，不能有参数，所以只能调用无参的构造函数，初始化对象

20. 已知有20个元素int类型向量V1，若用V1初始化为V2向量，语句是\_\_\_。

答案: (P151)vector <int>V2(V1);

〔解析〕 采用向量初始化另一个向量的形式：vector <type> name1(name);

三、改错题(本大题共5小题，每小题2分，共10分)下面的类定义中有一处错误，请用下

横线标出错误所在行并给出修改意见。

1. #include <iostream.h>

```
class Test
```

```
{private:
```

```
int x,y=20;
```

```
public:
```

```
Test(int i,int j){x=i,y=j;}
```

```
int getx(){return x;}
```



```
int gety(){return y;}

};

void main()

{Test mt(10,20);

cout<<mt.getx()<<endl;

cout<<mt.gety()<<endl;

}
```

答案：int x,y=20;在类内部不能对数据成员直接赋值。

【修改】int x,y;

2. #include <iostream.h>

```
class Test

{int x,y;

public:

fun(int i,int j)

{x=i;y=j;}

show()

{cout<<"x="<<x;

if(y)

cout<<"y="<<y<<endl;

cout<<endl;}

};

void main()

{Test a;

a.fun(1);

a.show();

a.fun(2,4);

a.show();

}
```

答案：int i,int j调用时，既有一个参数，也有两个参数，且没有重载，所以参数需要带默认值。所以int i,int j错误。

【修改】int i,int j=0//注j只要有一个int类型的数据就行。

3. #include <iostream.h>

```
class A

{int i;
```

```

public:

virtual void fun()=0;

A(int a)

{i=a;}

};

class B:public A

{int j;

public:

void fun()

{cout<<"B::fun() \n"; }

B(int m,int n=0):A(m),j(n){}

};

void main()

{A *pa;

B b(7);

pa=&b;

}

```

答案：B(int m,int n=0):A(m),j(n){}因为基类是抽象类，不能被实例化，所以在派生类中不能调用初始化基类对象。所以B(int m,int n=0):A(m),j(n){}错误，删去A(m)。

[修改] B(int m,int n=0):j(n){}

4. #include <iostream.h>

```

class X

{public:

int x;

public:

X(int x)

{cout<<this->x<<endl;}

X(X&t)

{x=t.x;

cout<<t.x<<endl;

}

void fun(X);

};

void fun(X t)

```

```
{cout<<t.x<<endl;}
```

```
void main()
```

```
{fun(X(10));}
```

答案: cout<<this->x=x<<endl;要输出this->x=x表达式的值要加括号。

[修改] cout<< (this->x=x) <<endl;

5. #include <iostream.h>

```
#include <string.h>
```

```
class Bas
```

```
{public:
```

```
Bas(char *s="\0"){strcpy(name,s);}
```

```
void show();
```

```
protected:
```

```
char name [20] ;
```

```
};
```

```
Bas b;
```

```
void show()
```

```
{cout<<"name:"<<b.name<<endl;}
```

```
void main()
```

```
{Bas d2("hello");
```

```
show();
```

```
}
```

答案: void show();是普通函数不是成员函数, 但是要访问类成员, 需要定义为友元函数。

[修改] friend void show();

四、完成程序题(本大题共5小题, 每小题4分, 共20分)

1. 在下面程序横线处填上适当字句, 以使该程序执行结果为:

50 4 34 21 10

0 7.1 8.1 9.1 10.1 11.1

```
#include <iostream.h>
```

```
template <class T>
```

```
void f (_____)
```

```
{_____;
```

```
for (int i=0;i<n/2;i++)
```

```
t=a [i] , a [i] =a [n-1-i] , a [n-1-i] =t;
```

```
}
```

```

void main ()

{int a [5] ={10,21,34,4,50};

double d [6] ={11.1,10.1,9.1,8.1,7.1};

f(a,5);f(d,6);

for (int i=0;i<5;i++)

cout <<a [i] << "";

cout <<endl;

for (i=0;i<6;i++)

cout << d [i] << "";

cout << endl;

}

```

答案: T a [] ,int n,T t=0;

【解析】不同的数据类型的调用，使用了模板。f函数增加t变量，因为实参类型不同，所以t的类型应该是T类型的。

2. 在下面程序的底画线处填上适当的字句，使该程序执行结果为40。

```

#include <iostream.h>

class Test

{ public:

_____;

Test (int i=0)

{x=i+x;}

int Getnum()

{return Test::x+7;}

};

_____;

void main()

{Test test;

cout<<test.Getnum()<<endl;;

}

```

答案: static int x;,int Test::x=30;

【解析】从成员函数访问方式类名：：成员可知是静态成员所以static int x;从结果要对初始化为30，且在类外进行初始化， int Test::x=30;。

3. 在下列程序的空格处填上适当的字句，使输出为：0，2，10。

```

#include <iostream.h>

```

```

#include <math.h>

class Magic

{double x;

public:

Magic(double d=0.00):x(fabs(d))

{}

Magic operator+(_____)

{

return Magic(sqrt(x*x+c.x*c.x));

}

_____operator<<(ostream & stream,Magic & c)

{ stream<<c.x;

return stream;

}

};

void main()

{Magic ma;

cout<<ma<<" "<<Magic(2)<<" "<<ma+Magic(-6)+

Magic(-8)<<endl;

}

```

答案：operator+(Magic&c),friend ostream&operator

【解析】对加法进行重载，operator+(Magic & c)，是对插入符进行重载，要访问成员所以定义为友元函数，friend ostream & operator。

4. 下面是一个输入半径，输出其面积和周长的C++程序，在下划线处填上正确的语句。

```

#include <iostream>

_____;

_____;

void main()

{double rad;

cout<<"rad=";

cin>>rad;

double l=2.0*pi*rad;

double s=pi*rad*rad;

cout<<"\n The long is: "<<l<<endl;

```

```
cout<<"The area is: "<<s<<endl;}
```

答案: using namespace std,#define pi 3.14159

【解析】进行输入或输出要引入iostream, 所以using namespace std;从标点看没有分号, 所以使用宏定义, #define pi 3.14159。

5. 程序实现大写字母转换成小写字母。

```
#include <iostream.h>
```

```
void main()
```

```
{char a;
```

```
_____;
```

```
cin>>a;
```

```
if(_____)
```

```
a=a+i;
```

```
cout<<a<<endl;
```

```
}
```

答案: int i=32;; a>=A && a<=Z

【解析】大写字母变小写字母相差32, 需要对i声明并初始化。大写字母变小写字母。要判断字符是大写字母。

五、程序分析题(本大题共4小题, 每小题5分, 共20分)

1. 给出下面程序输出结果。

```
#include<iostream.h>
```

```
class a
```

```
{public:
```

```
virtual void print()
```

```
{cout<< "a prog..."<< endl;};
```

```
};
```

```
class b:public a
```

```
{};
```

```
class c:public b
```

```
{public:
```

```
void print(){cout<<"c prog..."<<endl;}
```

```
};
```

```
void show(a *p)
```

```
{(*p).print();
```

```
}
```

```

void main()

{a a;

b b;

c c;

show(&a);

show(&b);

show(&c);

}

```

答案: a prog...

a prog...

c prog...

【解析】考查多态性的。a类对象调用本身的虚函数，b类因为没有覆写print，所以仍然调用基类的虚函数。而c类重新定义print虚函数，所以调用c类的print。

2. 给出下面程序输出结果。

```

#include <math.h>

#include <iostream.h>

#include <iomanip.h>

bool fun(long n);

void main()

{long a=10,b=30,l=0;

if(a%2==0) a++;

for(long m=a;m<=b;m+=2)

if(fun(m))

{if(l++%10==0)

cout <<endl;

cout <<setw(5) <<m;

}

}

bool fun(long n)

{int sqrtm=(int)sqrt(n);

for(int i=2;i<=sqrtm;i++)

if(n%i==0)

return false;

return true;
}

```

```
}
```

答案: 11 13 17 19 23 29

[解析] 循环体用来判断n是否是质数的函数,在main函数判断10~30之间质数。

3. 给出下面程序输出结果。

```
#include <iostream.h>

class Test
{
    int x,y;
public:
    Test(int i,int j=0)
    {x=i;y=j;}

    int get(int i,int j)
    {return i+j;}
};

void main()
{
    Test t1 (2),t2(4,6);

    int (Test::*p)(int,int=10);

    p=Test::get;

    cout<<(t1.*p)(5)<<endl;

    Test *p1=&t2;

    cout<<(p1->*p)(7,20)<<endl;
}
```

答案: 15 27

[解析] 指向类成员函数的指针的使用, \*p指向Test类中有两个参数的函数的一个指针。

P=Test::get.这样p就和get发生了联系。(t1.\*p)(5)等价于调用一个参数的get函数。

4. #include <iostream.h>

```
#include <string.h>
```

```
#include <iomanip.h>
```

```
class student
```

```
{char name [8] ;
```

```
int deg;
```

```
char level [7] ;
```

```
friend class process; // 说明友元类
```

```
public:
```

```
student(char na [],int d)
```



```

{ strcpy(name,na);

deg=d;

}

};

class process

{ public:

void trans(student &s)

{int i=s.deg/10;

switch(i)

{case 9:

strcpy(s.level, "优");break;

case 8:

strcpy(s.level,"良");break;

case 7:

strcpy(s.level,"中");break;

case 6:

strcpy(s.level,"及格");break;

default:

strcpy(s.level,"不及格");

}

}

void show(student &s)

{cout<<setw(10)<<s.name<<setw(4)<<s.deg<<setw(8)<<s.level<<endl;}

};

void main()

{ student st [] ={student("张三",78),student("李四",92),student("王五",62),student("孙六",88)};

process p;

cout<<"结 果:"<<"姓名"<<setw(6)<<"成绩"<<setw(8)<<"等级"<<endl;

for(int i=0;i<4;i++)

{ p.trans(st [i] );

p.show(st [i] );}

}

```

答案： 结果:姓名成绩等级

张三78中

李四92优

王五62及格

孙六88良

六、程序设计题(本大题共1小题，共10分)

7. 已定义一个Shape抽象类，在此基础上派生出矩形Rectangle和圆形Circle类，二者都有

GetPerim () 函数计算对象的周长，并编写测试main () 函数。

```
class Shape

{public:

Shape(){}

~Shape(){}

virtual float GetPerim()=0;

}
```

答案: class Rectangle:public Shape

```
{public:

Rectangle(float i,float j):L(i),W(j){}

~Rectangle(){}

float GetPerim(){return 2*(L+W);}

private:

float L,W;

};
```

```
class Circle:public Shape

{public:

Circle(float r):R(r){}

float GetPerim(){return 3.14*2*R;}

private:

float R;

};
```

```
void main()

{Shape * sp;

sp=new Circle(10);

cout<<sp->GetPerim ()<<endl;

sp=new Rectangle(6,4);

cout<<sp->GetPerim()<<endl;
```

