

C++程序设计试题及答案解析（三）

C++程序设计模拟试卷(三)

一、单项选择题

1. 设有定义int i;double j=5;; 则10+i+j值的数据类型是（）

- A. int
- B. double
- C. float
- D. 不确定

答案: B

解析: 考察数据的转换, j是double类型, 运算只能作同类型的运算, 所以要转换, 而int能自动转换为double类型, 所以结果是double类型。

2. 要禁止修改指针p本身, 又要禁止修改p所指向的数据, 这样的指针应定义为（）

- A. const char *p="ABCD";
- B. char *const p="ABCD";
- C. char const *p="ABCD";
- D. const char * const p="ABCD";

答案: D

解析: (P12)const char *p说明禁止通过p修改所指向的数据。char * const p则说明不能修改指针p的地址。因此const char * const p="ABCD"; 它禁止修改指针p本身, 又禁止修改p所指向的数据。

3. 类的构造函数被自动调用执行的情况是在定义该类的（）

- A. 成员函数时
- B. 数据成员时
- C. 对象时
- D. 友元函数时

答案: C

解析: (P75)建立对象时, 自动构造函数的初始化对象, 是系统自动调用的。而成员函数、友元函数, 需要用户直接调用, 因此选择C项。

4. 已知类A是类B的友元, 类B是类C的友元, 则（）

- A. 类A一定是类C的友元
- B. 类C一定是类A的友元
- C. 类C的成员函数可以访问类B的对象的任何成员
- D. 类A的成员函数可以访问类B的对象的任何成员

答案: C

解析: (P105)友元说明方法如下:

friend?<类名>;//友元类类名

使用友元可以访问所有成员:

- (1)友元关系不能被继承。
- (2)友元关系是单向的, 不具有交换性。所以, B项和D项错误。
- (3)友元关系不具有传递性。所以, A项错误。

5. 假定一个类的构造函数为“A(int i=4, int j=0) {a=i;b=j;}”, 则执行“A x (1);”语

句后, x.a和x.b的值分别为（）

- A. 1和0

B. 1和4

C. 4和0

D. 4和1

答案: A

解析: (P75)带默认的构造函数, 对应实参没有值时就采用形参值。调用构造函数时, i=1,不采用默认值, 而只有一个参数, j采用默认值0即j=0,因此a=1,b=0,选择A项。

6. 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。

7. 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。

8. 友元关系不能 ()

A. 提高程序的运行效率

B. 是类与类的关系

C. 是一个类的成员函数与另一个类的关系

D. 继承

答案: D

解析: (P111)友元可以是函数与类的关系即友元函数, 也可以类与类的关系即友元类, 但友元不能继承, 是单向性, 且不具有传递性。友元可以访问类中所有成员, 提高了访问的方便性。因此选择D项。

9. 语句ofstream f("TEMP.DAT",ios::app | ios::binary)?的功能是建立流对象f, 试图打

开文件TEMP.DAT 并与之连接, 并且 ()

A. 若文件存在, 将文件写指针定位于文件尾; 若文件不存在, 建立一个新文件

B. 若文件存在, 将其置为空文件; 若文件不存在, 打开失败

C. 若文件存在, 将文件写指针定位于文件首; 若文件不存在, 建立一个新文件

D. 若文件存在, 打开失败; 若文件不存在, 建立一个新文件

答案: A

解析: (P199)ios::binary, 采用二进制形式, ios::app定位到文件尾部。

10. 构造函数不具备的特征是 ()

A. 构造函数的函数名与类名相同

B. 构造函数可以重载

C. 构造函数可以设置默认参数

D. 构造函数必须指定类型说明

答案: D

解析: (P75)构造函数无返回类型不能继承但可以重载, 所以选择D项。

11. 在公有继承的情况下, 基类的公有或保护成员在派生类中的访问权限 ()

- A. 受限制
- B. 保持不变
- C. 受保护
- D. 不受保护

答案: B

解析: (P132)继承方式的不同派生类成员的权限也不同, 采用公有继承, 除了私有无法访问外, 公有、保护在派生类中保持不变, 所以选择B项。

12. 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。

13. C++对C语言做了很多改进, 即从面向过程变成为面向对象的主要原因是 ()

- A. 增加了一些新的运算符
- B. 允许函数重载, 并允许设置缺省参数
- C. 规定函数说明符必须用原型
- D. 引进了类和对象的概念

答案: D

解析: (P29)C++是一面向对象的语言, 面向对象的特征, 抽象、多态、继承和封装。

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. C++语言中所有在函数中定义的变量, 连同形式参数, 都属于 ()

- A. 全局变量
- B. 局部变量
- C. 静态变量
- D. 函数

答案: B

解析: 变量存储类可分为两类: 全局变量和局部变量。

(1) 全局变量: 在函数外部定义的变量称为全局变量, 其作用域为: 从定义变量的位置开始到源程序结束。使用全局变量降低了程序的可理解性, 软件工程学提倡尽量避免使用全局变量。

(2) 局部变量: 在函数内部定义的变量称为局部变量, 其作用域为: 从定义变量的位置开始到

函数结束。局部变量包含自动变量（auto）静态变量（static）以及函数参数。形参不能是静态的。所以选择B项。

16. 在私有继承的情况下，基类成员在派生类中的访问权限（）

- A. 受限制
- B. 保持不变
- C. 受保护
- D. 不受保护

答案：A

解析：(P132)私有继承下，基类中的公有或保护成员在派生类中也是私有的，所以选择A选项。

17. 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。

18. C++的继承性允许派生类继承基类的（）

- A. 部分特性，并允许增加新的特性或重定义基类的特性
- B. 部分特性，但不允许增加新的特性或重定义基类的特性
- C. 所有特性，并允许增加新的特性或重定义基类的特性
- D. 所有特性，但不允许增加新的特性或重定义基类的特性

答案：A

解析：(P129)派生类有两类成员：一是基类，二是自身类。派生类中的成员不能访问基类中的私有成员，可以访问基类中的公有成员和保护成员。

19. 对于int *pa [5] ;的描述，正确的是（）

- A. pa是一个指向数组的指针，所指向的数组是5个int型元素
- B. pa是一个指向某个数组中第5个元素的指针，该元素是int型变量
- C. pa [5] 表示某个数组的第5个元素的值
- D. pa是一个具有5个元素的指针数组，每个元素是一个int型指针

答案：D

解析：(P117)指针数组：数组元素都是相同类型的指针，相同类型的指针是说指针所指向的对象类型是相同的。例如，语句int *pa [5] ;定义了一个指针数组。在指针数组的定义中有两个运算符：*和 [] ，运算符 [] 的优先级高于*，所以*pa [5] 等价于*(pa [5])，pa [5] 表示一个数组，而*表示后面的对象为指针变量，合在一起*pa [5] 表示一个指针数组。该数组包含5个元素，每个元素都是指向int型的指针。所以选择D选项。

20. 以下基类中的成员函数表示纯虚函数的是（）

- A. virtual void tt()=0
- B. void tt(int)=0
- C. virtual void tt(int)
- D. virtual void tt(int){}

答案：A

解析：(P173)当在基类中不能为虚函数给出一个有意义的实现时，可以将其声明为纯虚函数，实现由派生类完成。格式：virtual<函数返回类型说明符><函数名>(<参数表>)=0;。

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. 单目运算符作为类成员函数重载时，形参数为___个。

答案：(P189)0

【解析】单目运算符使用成员函数重载可以不用形参，双目运算符使用一个参数。

2. 抽象类中至少要有一个___函数。

答案：(P173)纯虚

【解析】至少有一个纯虚函数的类就称为抽象类，即不能实例化。

3. 设类A有成员函数void f(void); 若要定义一个指向类成员函数的指针变量pf来指向f，该

指针变量的声明语句是：___。

答案：(P117)void (A::*pf)(void)=&A::f;

【解析】void (A::*pf)(void)=&A::f;指向成员函数的指针，它相当于两条语句

：void (A::*pf)(void);和pf=&A::f;。

4. 执行下列程序

```
double a=3.1415926,b=3.14;
```

```
cout<<setprecision(5)<<a<<" "<<setprecision(5)<<b<<endl;
```

程序的输出结果是___。

答案：3.1416, 3.14

【解析】题目设置精度即有效数字都是5，a四舍五入是3.1416，b是3.14。

5. vector类中用于删除向量中的所有对象的方法是___。

答案：(P151)clear()

【解析】向量的使用。返回向量中对象的方法有：front()back () operator []，在向量中删

除对象的方法pop_back erase clear。

6. 重载的运算符保持其原有的___、优先级和结合性不变。

答案：(P183)操作数

【解析】运算符重载时要遵循以下规则：

(1)除了类属关系运算符“.”、成员指针运算符“.”、作用域运算符“::”、sizeof运算符

和三目运算符“?:”以外，C++中的所有运算符都可以重载。

(2)重载运算符限制在C++语言中已有的运算符范围内的允许重载的运算符之中，不能创建新的

运算符。

(3)重载之后的运算符不能改变运算符的优先级和结合性，也不能改变运算符操作数的个数及

语法结构。

7. 编译时的多态性通过___函数实现。

答案：(P165)重载

【解析】编译多态性，实现的方法主要通过函数的重载或运算符的重载。

8. 基类的公有成员在派生类中的访问权限由___决定。

答案：(P132)访问控制方式或继承方式

9. 假设类X的对象x是类Y的成员对象，则“Y Obj”语句执行时，先调用类___的构造函数。

答案：(P130)X

【解析】派生类中的构造函数的执行顺序，先基类后派生类。

10. 下列程序段的输出结果是___。

```
cout.setf(ios::showpos);
```

```
cout<<509.3<<endl;
```

答案：(P193)+509.3

【解析】输入、输出格式ios::showpos用于输出数据的符号位。

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. C + + 中ostream的直接基类_____。

答案：(P193)ios

【解析】istream和ostream的直接基类是ios。

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. C++中有两种继承:单继承和_____。

答案：(P138)多继承

【解析】单继承和多继承,多继承即有多个基类。

15. 在C++中,利用向量类模板定义一个具有10个int的向量A,其元素均被置为1,实现此操作的语句是_____。

答案：(P151)vector<int>A(10,1)

【解析】定义向量列表vector<int>A(10,1),使用两个参数,10表示长度,1表示数值。

16. vector类中向向量尾部插入一个对象的方法是_____。

答案：(P157)push_back

17. C++语言中如果调用函数时,需要改变实参或者返回多个值,应该采取_____方式。

答案：(P51)传地址或引用

【解析】传地址即指针,在函数中通过指针修改它指向的变量的值时,实参也就变化了。使用引用,直接修改变量的别名即引用的值,该变量也就随着变化。

18. 若函数的定义处于调用它的函数之前,则在程序开始可以省去该函数的_____语句。

答案：声明

【解析】函数使用有两部分:声明和定义。定义在前,可以无声明;但函数定义在后,调用在前的话,需要先声明函数的原型。

19. 在C++中有两种参数传递方式:传值和_____。

答案：(P51)传引用

【解析】(1)传值调用又分为数据传值调用和地址传值调用。(2)引用调用是将实参变量值传递给形参,而形参是实参变量的引用名。引用是给一个已有变量起的别名,对引用的操作就是对该引用变量的操作。

20. 将指向对象的引用作为函数的形参,形参是对象的引用,实参是_____。

答案：(P53)对象名

【解析】实参与形参类型要一致,形参是对象的引用,实参应该是对象名。

三、改错题(本大题共5小题,每小题4分,共20分)

1. class ABC

```
{int a;
```

```
public:
```

```
ABC(int aa)a(aa){}
```

```
};
```

答案：ABC(int aa)a(aa){}初始化列表格式错误。

[修改] ABC(int aa): a(aa){}

2. #include <iostream.h>

```
class Aton
```

```
{int X,Y;
```

```
protected:
```

```
int zx,zy;
```

```
public:
```

```
void init(int i,int j){zx=i;zy=j;}
```

```
Aton(int i,int j,int n=0,int m=0)
```

```
{X=i,Y=j,zx=m,zy=n;
```

```
}
```

```
};
```

```
void main()
```

```
{Aton A(25,20,3,5);
```

```
A.init(5,9);
```

```
cout<<A.X()<<endl;
```

答案：int X,Y;因为X,Y都是私有的，在类外无法直接访问。

[修改] public: int X,Y;

3. #include <iostream.h>

```
class Bas
```

```
{public:
```

```
~Bas(){cout<<"Bas construct"<<endl;}
```

```
virtual void f()=0;
```

```
};
```

```
class Dev:public Bas
```

```
{public:
```

```
~Dev(){cout<<"Bas construct"<<endl;}
```

```
virtual void f(){cout<<"Dev::f"<<endl;}
```

```
};
```

```
void main()
```

```
{Bas *a=new Bas();
```

```
Dev p;
```

```
a=&p;
```

```
a->f();
```

```
}
```

答案： [修改] Bas *a;

[解析] Bas *a=new Bas();抽象类不能被实例化，但可以声明指针或引用，所以不能用new，因为new产生临时对象。

4. 以下程序实现交换a,b变量的值，请用下横线标出错误所在行并给出修改意见。

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

```
void swap(int &a,int &b)
```

```
{a=a+b;
```

```

b=a-b;

a=a-b;

}

void main()

{int a=19,b=15;

cout<<"a="<<a<<" ,b="<<b<<endl;

swap(&a,&b);

cout<<"a="<<a<<" ,b="<<b<<endl;

}

```

答案： swap(&a,&b);函数的形参是变量的引用，调用时的实参应该是地址。

[修改] swap(a, b);

5. #include <iostream.h>

```

void main()

{int i(3),j(8);

int * const p=&i;

cout<<*p<<endl;

p=&j;

cout<<*p<<endl;

}

```

答案： int * const p=&i;在指针变量前加const表示一个常指针即地址不能变化，它指向的变量不能改变且定义时必须设置指向变量或对象的地址。

[修改] int *p=&i;

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内容，使程序执行结果为："hello, andylin"。

```

#include <iostream>

#include <string.h>

using namespace std;

class mystring

{public:

char * pdata;

mystring(int len)

{pdata=new char [len+1] ;

}

~mystring()

{delete pdata;}

void show(){cout<<pdata<<endl;}

};

void fun(mystring** array,int len)

{mystring*old=*array;

_____ ;

memcpy(*array, old, len);

}

void main()

{mystring str(20);

```



```

mystring* pstr=&str;

mystring** ppstr=&pstr;

strcpy(str.pdata,"hello,andylin");

fun(ppstr, 20);

_____

}

```

答案： *array=new mystring(len);, (**ppstr).show();或str.show();

【解析】调用mystring类的构造函数开辟空间，后进行字符的复制。输出可以直接使用str或者使用二级指针。

2. 在下面程序横线处填上适当字句，完成类的定义。

```

class line;

class box

{ private:

int color;

int upx, upy;

int lowx, lowy;

public:

_____

void set_color (int c){color=c;}

void define_box (int x1, int y1, int x2, int y2)

{upx=x1;upy=y1;lowx=x2;lowy=y2;}

};

class line

{ private:

int color;

int startx, starty;

int endx, endy;

public:

friend int same_color(line l,box b);

void set_color (int c) {color=c;}

void define_line (_____)

{startx=x1;starty=y1;endx=x2;endy=y2;}

};

int same_color(line l, box b)

{if (l.color==b.color) return 1;

return 0;

}

```

答案： friend int same_color(line l, box b);,int x1, int y1, int x2, int y2

【解析】成员函数作为友元函数的使用。使用friend关键字。由函数体可知形参的类型和个数。

3. 下面程序用来求直角三角形斜边长度。

```

#include <iostream.h>

#include <math.h>

class Point

{private:

```

```

double x,y;

_____

public:

Point(double i=0,double j=0)

{x=i;y=j;}

Point(Point &p)

{x=p.x;y=p.y;}

};

class Line

{private:

Point p1,p2;

public:

Line(Point &xp1,Point &xp2):_____{}

double GetLength();

};

double Line::GetLength()

{double dx=p2.x-p1.x;

double dy=p2.y-p1.y;

return sqrt(dx*dx+dy*dy);

}

void main()

{ Point p1,p2(6,8);

Line L1(p1,p2);

cout<<L1.GetLength()<<endl;

}

```

答案： friend Line;,p1(xp1),p2(xp2)

【解析】友元类的使用，定义Line是Point类的友元类，成员对象的初始化采用列表的形式。

4. 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;。

5. 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。

五、程序分析题(本大题共2小题，每小题5分,共10分)

1. 运行程序，写出程序执行的结果。

```
#include <iostream.h>

void main()

{int a,b,c;

char ch;

cin>>a>>ch>>b>>c;//从键盘上输入1.5×c×10×20,*表示一个空格

cout<<a<<endl<<ch<<endl<<b<<endl<<c<<endl;

}
```

答案： 1

.

5

0

【解析】使用cin进行输入字符的输入的问题。1-->a,.-->ch,5-->b,空格转换为零给了c。

2. 给出下面程序输出结果。

```
#include <iostream.h>

class A

{public:
```

```

A()

{cout<<"As cons."<<endl;}

virtual ~A()

{cout<<"As des."<<endl;}

virtual void f()

{cout<<"As f()."<<endl;}

void g()

{f();}

};

class B:public A

{public:

B()

{f();cout<<"Bs cons."<<endl;}

~B()

{cout<<"Bs des."<<endl;}

};

class C:public B

{public:

C()

{cout<<"Cs cons."<<endl;}

~C()

{cout<<"Cs des."<<endl;}

void f()

{cout<<"Cs f()."<<endl;}

};

void main()

{A *a=new C;

a->g();

delete a;

}

```

答案：As f().

Bs cons.

Cs cons.

Cs f().

Cs des.

Bs des.

As des.

【解析】定义C类对象时要调用基类构造函数从A到B再到C，调用B的构造函数时，B类没有f（），则指向来自A类的f（）函数。同时用基类的指针指向了派生类对象。最后析构函数的执行。

六、程序设计题（本大题共1小题，共10分）

1. 已知交通工具类定义如下。

要求：（1）实现这个类；（2）定义并实现一个小车类car，是它的公有派生类，小车本身的私有属性有载人数，小车的函数有init(设置车轮数，重量和载人数)，getpassenger(获取载人数

), print (打印车轮数, 重量和载人数) 。

```
class vehicle

{protected:

int wheels;//车轮数

float weight;//重量

public:

void init(int wheels,float weight);

int get_wheels();

float get_weight();

void print();

};

void vehicle::init(int wheels,float weight)

{this->wheels=wheels;

this->weight=weight;

cout<<wheels<<endl;

}

int vehicle::get_wheels()

{return wheels;

}

float vehicle::get_weight()

{return weight;}

void vehicle::print()

{cout<<"车轮数: "<<wheels<<","<<"重量: "<<weight<<endl;}
```

答案: class car:public vehicle

```
{private:int passengers;

public:

void init(int wheels,float weight,int pass);

int getpassenger();

void print();};

void car::init(int wheels,float weight,int pass)

{vehicle::init(wheels,weight);

passengers=pass;}

int car::getpassenger()

{return passengers;}

void car::print()

{vehicle::print();

cout<<"可载人数: "<<passengers<<endl;

}
```

C++程序设计及答案解析 (四)