

# C++程序设计试题及答案解析（五）

C++程序设计模拟试卷(五)

## 一、单项选择题

1. 静态成员函数没有（）

- A. 返回值
- B. this指针
- C. 指针参数
- D. 返回类型

答案: B

解析: (P107)静态成员函数是普通的函数前加入static,它具有函数的所有的特征: 返回类型、形参, 所以使用(P107)静态成员函数, 指针可以作为形参, 也具有返回值。静态成员是类具有的属性, 不是对象的特征, 而this表示的是隐藏的对对象的指针, 因此静态成员函数没有this 指针。静态成员函数当在类外定义时, 要注意不能使用static关键字作为前缀。由于静态成员函数在类中只有一个拷贝(副本), 因此它访问对象的成员时要受到一些限制: 静态成员函数可以直接访问类中说明的静态成员, 但不能直接访问类中说明的非静态成员; 若要访问非静态成员时, 必须通过参数传递的方式得到相应的对象, 再通过对象来访问。

2. 在类的定义中, 用于为对象分配内存空间, 对类的数据成员进行初始化并执行其他内部管理操作的函数是（）

- A. 友元函数
- B. 虚函数
- C. 构造函数
- D. 析构函数

答案: C

解析: (P75)定义构造函数作用就是初始化对象, 而析构函数释放对象空间。虚函数用于完成多态性, 友元增加访问方便性。

3. 所有在函数中定义的变量, 都是（）

- A. 全局变量
- B. 局部变量
- C. 静态变量
- D. 寄存器变量

答案: B

解析: (P136)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。全局变量增加了函数之间数据联系的渠道, 全局变量作用域内的函数, 均可使用、修改该全局变量的值, 但是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。auto变量意味着变量的存储空间的分配与释放是自动进行的。说明符auto可以省略。函数中的局部变量存放在栈空间。在函数开始运行时, 局部变量被分配内存单元, 函数结束时, 局部变量释放内存单元。因此, 任两个函数中的局部变量可以同名, 因其占有不同的内存单元而不影响使用。这有利于实现软件开发的模块化。

static变量是定义在函数体内的变量, 存放在静态存储区, 不用栈空间存储, 其值并不随存

储空间的释放而消失。

4. 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 ( )

A. 3

B. 4

C. 5

D. 9

答案: B

解析: (P79)a(2)调用1次带参数的构造函数，b [3] 调用3次无参数的构造函数，指针没有给它分配空间，没有调用构造函数。所以共调用构造函数的次数为4。

5. 如果表达式++a中的“++”是作为成员函数重载的运算符，若采用运算符函数调用格式，则

可表示为 ( )

A. a.operator++(1)

B. operator++(a)

C. operator++(a,1)

D. a.operator++()

答案: D

解析: (P186)运算符的重载，前缀先让变量变化。调用 + + a, 等价于a.operator++();注意无参的形式。后缀的话a++,等价于a.operator(0),带形参，形参名可省。

6. 已知f1和f2是同一类的两个成员函数，但f1不能直接调用f2，这说明 ( )

A. f1和f2都是静态函数

B. f1不是静态函数，f2是静态函数

C. f1是静态函数，f2不是静态函数

D. f1和f2都不是静态函数

答案: C

解析: (P107)普通成员函数可以调用静态函数，相反静态函数不能调用普通成员函数，这与普通函数与常成员函数相同。因此选择C项。

7. 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。

8. 解决定义二义性问题的方法有 ( )

A. 只能使用作用域分辨运算符

B. 使用作用域分辨运算符或成员名限定

C. 使用作用域分辨运算符或虚基类

D. 使用成员名限定或赋值兼容规则

答案: B

解析: (P139)解决二义性问题主要要两种方法: (1)赋值兼容规则; (2)虚基类。

9. 在main函数中可以用p.a的形式访问派生类对象p的基类成员a，其中a是 ( )

- A. 私有继承的公有成员
- B. 公有继承的私有成员
- C. 公有继承的保护成员
- D. 公有继承的公有成员

答案: D

解析: (P132)公有成员可以在类外访问, 保护类型成员可以在派生类中访问, 但不能在类外访问, 在main函数中访问, 说明a是公有成员。只有公有继承时a才能是公有的, 所以D项正确。

10. 在C++中不返回任何类型的函数应该说明为 ( )

- A. int
- B. char
- C. void
- D. double

答案: C

解析: 无形参或无返回值都可以用void来声明, int char double分别是整型、字符型和实型。

11. 若Sample类中的一个成员函数说明如下:

void set(Sample& a), 则Sample& a的含义是 ( )

- A. 指向类Sample的名为a的指针
- B. a是类Sample的对象引用, 用来作函数Set ( ) 的形参
- C. 将a的地址赋给变量Set
- D. 变量Sample与a按位与的结果作为函数Set的参数

答案: B

解析: (P53)成员函数使用对象的引用作为形参。该函数的功能是将已知对象的所有数据成员的值拷贝给相应对象的所有数据成员, 不会建立临时对象, 这里是对象的引用所以选择B。

12. 要实现动态联编必须 ( )

- A. 通过成员名限定来调用虚函数
- B. 通过对象名来调用虚函数
- C. 通过派生类对象来调用虚函数
- D. 通过对象指针或引用来调用虚函数

答案: D

解析: (P170)通过基类指针或基类引用来调用虚函数实现动态多态性, 静态多态性通过重载来实现的。所以选择D项。

13. 在派生类中定义虚函数时, 可以与基类中相应的虚函数不同的是 ( )

- A. 参数类型
- B. 参数个数
- C. 函数名称
- D. 函数体

答案: D

解析: (P170)虚函数在基类和派生类, 具有相同的返回类型、形参类型和形参个数, 而函数体可以根据不同的派生类或基类实现不同的操作, 即不同函数体。

14. 实现两个相同类型数加法的函数模板的声明是 ( )

- A. add(T x,T y)
- B. T add(x,y)
- C. T add(T x,y)
- D. T add(T x,T y)

答案: D

解析: (P63)实现两个相同类型数加法结果应该和操作数具有相同类型。进行加法运算后结果也是和参数具有相同类型, 需要返回值。A无返回值时要用void,B形参无类型, C形参y没有类型, 所以选择D项。

15. 下列不是描述类的成员函数的是 ( )

- A. 构造函数
- B. 析构函数
- C. 友元函数
- D. 拷贝构造函数

答案: C

解析: (P109)友元函数虽然不是成员函数但是可以访问类所有成员。构造函数、析构函数和拷贝构造函数(复制构造函数)都是类的特殊函数用于对象的创建和撤销, 所以选择C项。

16. 继承机制的作用是 ( )

- A. 信息隐藏
- B. 数据封装
- C. 定义新类
- D. 数据抽象

答案: C

解析: (P40)面向对象设计中的类的特点: 抽象、封装、继承和多态等, 继承用于对类的扩展, 所以选择C项。

17. 已知: p是一个指向类A数据成员m的指针, A1是类A的一个对象。如果要给m赋值为5, 正确的是 ( )

- A. A1.p=5;
- B. A1->p=5;
- C. A1.\*p=5;
- D. \*A1.p=5;

答案: C

解析: (P118)A中p是指针即地址, 错误; B选项中A1不是指针不能使用指向运算符->, 错误; “\*”比“.”级别要高, 所以D选项\*A1.p=5相当于(\*A1).p=5; 错误。另外涉及到指向成员函数时注意以下几点:

指向成员函数的指针必须于其赋值的函数类型匹配的三个方面: (1)参数类型和个数; (2)返回类型; (3)它所属的类类型。

成员函数指针的声明: 指向short型的Screen类的成员的指针定义如下:

```
short Screen::* ps_Screen;
```

ps\_Screen可以用\_height的地址初始化如下: short Screen::\*ps\_Screen=&Screen::\_height;

类成员的指针必须总是通过特定的对象或指向改类型的对象的指针来访问。是通过使用两个指向成员操作符的指针(针对类对象和引用的\*, 以及针对指向类对象的指针的->\*)。

18. 如果采用动态多态性, 要调用虚函数的是 ( )

- A. 基类对象指针
- B. 对象名
- C. 基类对象
- D. 派生类名

答案: A

解析: (P171)基类指针或者基类的引用调用虚函数都会产生动态多态性

19. 若有以下定义，则说法错误的是（）

```
int a=100,*p=&a;
```

- A. 声明变量p，其中\*表示p是一个指针变量
- B. 变量p经初始化，获得变量a的地址
- C. 变量p只可以指向一个整型变量
- D. 变量p的值为100

答案：D

解析：指针变量如同其他变量一样，在使用之前必须先声明。声明指针变量的格式为：

<类型名>\*<变量名>;

其中，<类型名>是指针变量所指向对象的类型，它可以是C++语言预定义的类型，也可以是用户自定义类型。<变量名>是用户自定义的标识符。符号\*表示<变量>是指针变量。而不是普通变量。  
。 \*表示指针，p是变量，p指向一个整型的变量，值为a的地址值，\*p=100。

20. C++语言建立类族是通过（）

- A. 类的嵌套
- B. 类的继承
- C. 虚函数
- D. 抽象类

答案：B

解析：(P174)类族即同一个类派生出来的类，各个类是兄弟或父子关系。

二、填空题（本大题共20小题，每小题1分，共20分）请在每小空的空格中填上正确答案。错填、不填均无分。

1. 假设int a=1,b=2;则表达式(++a/b)\*b--的值为\_\_\_。

答案：2

【解析】前缀++或——表示先使变量值变化，再使用，这和后缀恰恰相反。但是编译cout<<(++a/b)\*b--时，先++a/b值为1，后1\*b--，先取b=2，结果为2，再让b=1。

2. 复制构造函数使用\_\_\_作为形式参数。

答案：(P80)对象的引用

【解析】复制构造函数使用对象的引用来初始化一个新对象,避免临时产生对象。

3. 通过C++语言中的\_\_\_机制，可以从现存类中构建其子类。

答案：(P127)继承

【解析】继承概念，从现有的类生成新类，原有的类称为父类或基类，新类又称子类或派生类或衍生类，它是对基类的扩充。

4. 静态成员函数、友元函数、构造函数和析构函数中，不属于成员函数的是\_\_\_。

答案：(P109)友元函数

【解析】友元函数不是类成员，但可以访问类成员。类的封装性保证了数据的安全，但引入友元，虽然访问类是方便了，但确实破坏类访问的安全性。

5. 在下面的类定义中，私有成员有\_\_\_。

```
class Location
{
    int X,Y;

protected:

    int zeroX,zerxY;

    int SetZero(intzeroX, intzeroY);

private:

    int length,height;
```

```
public:

void init(int initX,int initY);

int GetX();

int GetY();

};
```

答案: (P69)X,Y,length,height

6. 在C++程序设计中，建立继承关系倒挂的树应使用\_\_\_继承。

答案: (P138)单

【解析】一个基类可以派生多个子类，一个子类可以再派生出多个子类，这样就形成了一个倒立的树。

7. C++支持的两种多态性分别是\_\_\_多态性和运行多态性。

答案: (P165)静态或编译

【解析】多态性包括静态（编译时）的和动态（运行时）的动态性。

8. C++中语句const char \* const p="hello"; 所定义的指针p和它所指的内容都不能被\_\_\_。

答案: (P12)修改

【解析】使用const修改的内容不能修改，这里同时修饰地址和值，表示地址和值都不变。

9. 在C++中，定义虚函数的关键字是\_\_\_。

答案: (P170)virtual

【解析】在普通函数前面用virtual修饰的函数，就称为虚函数。

10. 采用私有派生方式，基类的public成员在私有派生类中是\_\_\_成员。

答案: (P132)私有

11. 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。

12. 在C++中有两种参数传递方式即值传递和\_\_\_传递。

答案: (P51)引用

【解析】函数参数传递有传值和传引用两种。

13. 预处理命令以\_\_\_符号开头。

答案: (P183)operator

【解析】文件包含、预处理和编译都是以#开头。

14. 在构造函数和析构函数中调用虚函数时采用\_\_\_。

答案: (P167)静态联编

【解析】在析构或构造函数调用虚函数发生静态多态性。

15. C++是通过引用运算符\_\_\_来定义一个引用的。

答案: (P10)&

【解析】引用是C不具有使用方法，它表示变量的别名，在函数中使用很频繁，因为调用形式同传值调用，但修改形参实参也会相应改变的特征。

16. 如果要把类B的成员函数void fun()说明为类A的友元函数，则应在类A中加入语句\_\_\_。

答案: (P111)friend void B::fun();

【解析】声明成员函数作为另外一个类的友元函数时，使用类作用域运算符：：。

17. 如果要把PI声明为值为3.14159类型为双精度实数的符号常量，该声明语句是\_\_\_。

答案: (P6)const double PI(3.14159);或者const double PI = 3.14159;

【解析】使用const声明符号常量，常量和常量值可以用括号也可以赋值号。

18. 在C++四个流对象中，\_\_\_\_用于标准屏幕输出。

答案：(P194)cout

【解析】cin、cout、cerr和clog中cin用于输入，cout用于输出，cerr、clog错误处理。

19. 执行下列代码

```
int a=32;

double c=32;

cout.setf(ios::hex);

cout<<"hex:a="<<a<<" ,c="<<c<<endl;

cout.unsetf(ios::hex);

程序的输出结果为____。
```

答案：(P193)hex:a=20,c=32

【解析】用十六进制只能输出整型数据，而不能将其它类型数据转换成十六进制的数据输出。所以double类型不变仍然是32（double类型）。

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4分，共20分）

1. #include <iostream.h>

```
class A
{ private:
int x;
public:
A(int i){x=i;}
A(){x=0;}
friend int min(A&,A&);
};

int min(A &a,A &b)
{ return (a.x>b.x)?a.x:b.x;
}

void main()
{ A a(3),b(5);
cout<<a.min(a,b)<<endl;
}
```

答案：cout<<a.min(a,b)<<endl;友元函数不是类成员，所以对象a不能使用a.min(a,b)这种方法。min就是一个普通的友元函数。

【修改】cout<<min(a,b)<<endl;

2. #include <iostream.h>

```
class shape
{public:
virtual int area(){return 0;}
};

class rectangle:public shape
{public:
```

```

int a, b;

void setLength (int x, int y) {a=x;b=y;}

int area() {return a*b;}

};

void main()

{rectangle r;

r.setLength(3,5);

shape s1,*s2=&r;

cout <<r.area() <<endl;

s2=s1;

cout <<s2.area()<<endl;

}

```

答案： shape s1,\*s2=r;指针使用错误。s是指针使用它指向对象的成员有两种方法，有下面两行可知，使用的是引用。

[\[修改\]](#) 改为shape &s=r;

3. 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。

```

#include <iostream.h>

template <class T>

class A

{private:

T x,y,s;

public:

A(T a,T b)

{x=a,y=b;s=x+y;}

void show()

{cout<<"x+y="<<s<<endl;

}

};

void main()

{ A <int>add(10,100);

add.show();

}

```

答案： [\[修改\]](#) A <int>add(10,100);

[\[解析\]](#) A add(10,100);类模板的使用，参数实例化后生成模板类。用类模板定义对象时要指定参数类型。

4. 生成具有n个元素的动态数组。

```

#include <iostream.h>

void main()

{int n;

cin>>n;

int a [n] ;

a [0] =2;

cout<<a [0] <<endl;

}

```



答案: int a [n] ;生成具有n个元素的动态数组, 要使用new, 所以int a [n] ;错误。

[修改] int \*a=new int [n] ;

5. #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){}

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函数增加变量，因为实参类型不同，所以t的类型应该是T类型的。

2. 完成下面类中成员函数的定义。

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

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

```
class Arr
```

```
{protected:
```

```
float *p;
```

```
int n;//数组大小 (元素个数)
```

```
public:
```

```
Arr(int sz=10)
```

```
{ n=sz;
```

```
p=new float [n] ;
```

```
}
```

```
~Arr(void)
```

```
{
```

```
_____
```

```
}
```

```
int Getn(void) const
```

```
{
```

```
return n;
```

```
}
```

```
float & operator [] (int i)
```

```
{
```

```
_____
```

```
}
```

```
void Print();
```

```
};
```

```
void Arr::Print()
```

```
{int i;
```

```
for(i=0;i< this->Getn();i++)
```

```
{if (i%10==0)
```

```
cout << endl;
```

```
cout<<setw(6)<<p [i] ;
```

```
}
```

```
cout<<endl;
```

```
}
```

```
void main()
```

```
{Arr a(20);
```

```
for (int i=0;i<a.Getn();i++)
```

```

a [i] =i* 2;

a.Print();

}

```

答案： delete p;， return p [i] ；

【解析】在析构函数中释放对象空间。第二个是对 [] 运算符的重载，函数返回类型是实型，形参i，取得下标为i的元素的值。

3. 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: "<<|<<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。

4. 在下划线处填上缺少的部分。

```

#include <iostream.h>

class Samp

{public:

void Setij(int a,int b){i=a,j=b;}

~Samp()

{cout<<"Destroying.."<<|<<endl;}

int GetMuti(){return i*j;}

protected:

int i;

int j;

};

int main()

{Samp *p;

p=new Samp [5] ;

if(!p)

{cout<<"Allocation error \n";

return 1;

}

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

p [j] .Setij(j,j);

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

cout<<"Muti ["<<k<<"] is:"<<p [k] ._____<<endl;

```

---

```
return 0;
```

```
}
```

答案: GetMuti(), delete [] p;

[解析] 调用只有一个有返回值的成员函数, 释放对象数组所占的空间。

5. 请在下面程序的横线处填上适当内容, 以使程序完整,并使程序的输出为:

11,10

13,12

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

```
class A
```

```
{int a;
```

```
public:
```

```
A(int i=0){a=i;}
```

```
int Geta(){return a;}
```

```
void show(){cout<<a<<endl;}
```

```
};
```

```
class B
```

```
{A a;
```

```
int b;
```

```
public:
```

```
B(int i,int j)_____
```

```
{}
```

```
void show(){cout<<a.Geta()<<","<<b<<endl;}
```

```
};
```

```
void main()
```

```
{B b [2] =(B(10,11),B(12,13));
```

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

```
_____
```

```
}
```

答案: :a(j),b(i), b [i] .show();

[解析] 在构造函数中对数据成员初始化, 从结果先输出a, 后b, 所以对a = j, b=i;在循环中

输出成员, 调用show成员。

五、程序分析题(本大题共2小题, 每小题5分,共10分)

1. 给出下面程序输出结果。

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

```
class Base
```

```
{private:
```

```
int Y;
```

```
public:
```

```
Base(int y=0) {Y=y;cout<<"Base("<<y<<") \ n";}
```

```
~Base() {cout<<"~Base() \ n";}
```

```
void print() {cout <<Y<< "";} 
```

```
};
```

```
class Derived:public Base
```

```

{private:

int Z;

public:

Derived (int y, int z):Base(y)

{Z=z;

cout<<"Derived("<<y<<","<<z<<") \ n";

}

~Derived() {cout<<" ~ Derived() \ n";}

void print()

{Base::print();

cout<<Z<<endl;

}

};

void main()

{Derived d(10,20);

d.print();

}

```

答案： Base(10)

Derived(10,20)

10 20

~ Derived()

~Base()

【解析】派生类对象，先调用基类构造函数输出Base(10)，后调用派生类构造函数输出Derived(10,20)，后执行d.print(),调用派生类的print，再调用Base::print()输出10，后返回输出z的值20。后派生类析构，再基类析构。

2. 给出下面程序输出结果。

```

#include <iostream.h>

class test

{int x;

public:

test(int i=0):x(i){}

virtual void fun1()

{cout << "test::x"<<x<<endl;}

};

class ft:public test

{int y;

public:

void fun1(){cout <<"ft::y="<<y<<endl;}

ft(int i=2):test(i),y(i){}

};

void main()

{ ft ft1(3);

void (test::*p)();

p=test::fun1;

```

```
(ft1.*p)();  
}
```

```
int student::getdeg()
```

```
{return deg;}__
```

## C++程序设计模拟试卷(六)

### 一、单项选择题

1. 当一个类的某个函数被说明为virtual时，该函数在该类的所有派生类中（）

- A. 都是虚函数
- B. 只有被重新说明时才是虚函数
- C. 只有被重新说明为virtual时才是虚函数
- D. 都不是虚函数

答案：A

解析：(P170)在基类声明为virtual的函数为虚函数，在派生类中只要有相同的函数（函数名相同、返回值相同、形参类型和个数相同）即使不用virtual说明，也都是虚函数。

2. 以下基类中的成员函数表示纯虚函数的是（）

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

答案：C

解析：(P173)纯虚函数是特殊的虚函数，没有函数体，形式为：virtual返回类型函数名（形参列表）=0；因此选C项。

3. 下面对静态数据成员的描述中，正确的是（）

- A. 静态数据成员可以在类体内进行初始化
- B. 静态数据成员可以直接用类名或者对象名来调用
- C. 静态数据成员不能用private控制符修饰
- D. 静态数据成员不可以被类的对象调用

答案：B

解析：(P107)静态成员可用类名或者对象名来调用，静态数据成员必须在类外进行初始化。静态成员可以用public、private和protected修饰。所以选B项。

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. 类B是类A的公有派生类，类A和类B中都定义了虚函数func(),p是一个指向类A对象的指针，则p->A::func()将（）

- A. 调用类A中的函数func()
- B. 调用类B中的函数func()
- C. 根据p所指的對象类型而确定调用类A中或类B中的函数func()

D. 既调用类A中函数，也调用类B中的函数

答案: A

解析: (P117)指向类成员指针的使用, A::func()是明确调用A类的func函数, 所以不管p指向基类或者派生类对象, 都执行基类虚函数。注意p->A::func()和p->fun();进行区分。如果使用p->fun(), 因为p指向派生类对象, 由动态多态性可知要调用派生类的虚函数。

6. 在面向对象的程序设计中, 首先在问题域中识别出若干个 ( )

A. 函数

B. 类

C. 文件

D. 过程

答案: B

解析: (P31)面向过程的和面向对象都具有、函数、文件和过程这些概念, 而面向对象程序才有类和对象的特征。所以选择B。

7. 在下列成对的表达式中, 运算结果类型相同的一对是 ( )

A.  $7.0 / 2.0$ 和 $7.0 / 2$

B.  $5 / 2.0$ 和 $5 / 2$

C.  $7.0 / 2$ 和 $7 / 2$

D.  $8 / 2$ 和 $6.0 / 2.0$

答案: A

解析: 小数默认的类型为double类型, 整数除以整数结果是取整的结果。A、B、C和D项的第一项分别是double、double、double和int类型的, 第二项分别是double、int、int和double类型, 所以选择A项。

8. 下列不具有访问权限属性的是 ( )

A. 非类成员

B. 类成员

C. 数据成员

D. 函数成员

答案: A

解析: 类成员包括成员函数和数据成员, 都可以使用访问权限public、private和protected来修饰, 而普通的变量不能使用访问权限来说明。

9. 以下有关继承的叙述正确的是 ( )

A. 构造函数和析构函数都能被继承

B. 派生类是基类的组合

C. 派生类对象除了能访问自己的成员以外, 不能访问基类中的所有成员

D. 基类的公有成员一定能被派生类的对象访问

答案: C

解析: (P129)构造函数和析构函数不能被派生类继承, A项错误。派生类是基类的扩展, B项错。派生类可以访问基类公有和保护类型的成员, 不能访问基类私有成员。D项基类是公有的成员, 若采用私有继承, 派生类对象不能直接访问。选择C项。

10. 下列有关模板和继承的叙述正确的是 ( )

A. 模板和继承都可以派生出一个类系

B. 从类系的成员看, 模板类系的成员比继承类系的成员较为稳定

C. 从动态性能看, 继承类系比模板类系具有更多的动态特性

D. 相同类模板的不同实例一般没有联系, 而派生类各种类之间有兄弟父子等关系



答案: D

解析: (P145)类是相同类型事物的抽象, 具有不同的操作。而模板是不同类型的事物, 具体相同的操作的抽象。类模板的实例化后, 各个对象没有任何关系。而类对象是通过派生、继承等关系的关系。

11. 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。

12. 要采用动态多态性, 说法正确的是 ( )

- A. 基类指针调用虚函数
- B. 派生类对象调用虚函数
- C. 基类对象调用虚函数
- D. 派生类指针调用虚函数

答案: A

解析: (P170)使用基类的指针或引用, 由指向或引用的对象来决定调用不同类的虚函数。所以选择A。

13. C++类体系中, 不能被派生类继承的有 ( )

- A. 转换函数
- B. 构造函数
- C. 虚函数
- D. 静态成员函数

答案: B

解析: (P127)构造函数不能被继承, 而转换函数、虚函数和静态成员函数都可以被继承, 所以选择B项。

14. 下列不是描述类的成员函数的是 ( )

- A. 构造函数
- B. 析构函数
- C. 友元函数
- D. 拷贝构造函数

答案: C

解析: (P109)构造函数、析构函数、拷贝构造函数都是特殊的成员函数, 友元则不是成员函数。所以选择C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. 下列不是描述类的成员函数的是 ( )

- A. 构造函数
- B. 析构函数
- C. 友元函数
- D. 拷贝构造函数

答案: C

解析: (P113)友元函数不是类成员, 只是它可以访问类中的成员。

17. 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—个实体
- C. 对象是类的—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。

18. 派生类的构造函数的成员初始化列表中, 不能包含 ( )

- 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

答案: C

解析: (P130)派生类的构造函数的成员初始化, 包括基类成员、基类对象成员和派生类成员。因为本身就是初始化定义的对象, 在构造函数中再进行该类对象的初始化产生了循环定义, 或者类中不能包括本身类的成员对象。所以选择C项。

19. 关于new运算符的下列描述中, 错误的是 ( )

- A. 它可以用来动态创建对象和对象数组
- B. 使用它创建的对象或对象数组可以使用运算符delete删除
- C. 使用它创建对象时要调用构造函数
- D. 使用它创建对象数组时必须指定初始值

答案: D

解析: (P78)new创建的对象数组不能指定初始值, 所以调用无参的构造函数, 选择D项。

20. 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. 4和5
- B. 4和20
- C. 4和24
- D. 20和5

答案: C

解析: (P75)执行构造函数将数据成员进行赋值, aa++是后加, 先赋值a = 4, ++bb,bb变量值先自加为6, 再与a相乘, 所以b = 24。

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. 在C++中, 编译指令都是以\_\_\_\_ (符号) 开始。

答案: #

C++的源程序可包括各种编译指令，以指示编译器对源代码进行编译之前先对其进行预处理。所有的编译指令都以#开始，每条指令单独占用一行，同一行不能有其他编译指令和C++语句（注释例外）

2. 在函数前面用\_\_\_\_保留字修饰时，则表示该函数表为内联函数。

答案: (P59)inline

【解析】内联函数，用来提高程序运行速度。在类内部定义的函数也是内联函数。

3. 执行下列程序

```
int a=21,j=16;

cout.setf(ios::hex);

cout<<a<<<"";

cout.unsetf(ios::hex);

cout<<j<<endl;
```

程序的输出结果是\_\_\_\_。

答案: (P196)1516

【解析】a=21以十六进制输出，j=16以十进制输出。

4. 在单继承和多继承方式中，面向对象的程序设计应尽量使用\_\_\_\_继承。

答案: (P138)单

【解析】多继承是单继承的扩展，且多继承易产生二义性等问题。

5. 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>

6. 在C++中，访问指针所指向的对象的成员使用\_\_\_\_运算符。

答案: ->或.

【解析】使用指针访问成员有两种方法：成员运算符“.”或指向运算符“->”。

7. 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。

8. 局部对象和全局对象中，在同一程序中\_\_\_\_生存期最长。

答案: 全局对象

变量或对象的生命期或者作用域的不同，全局对象生命期长。

9. this指针始终指向调用成员函数的\_\_\_\_。

答案: 对象

this指针是隐藏的指针，它指向调用函数的对象。

10. 派生类的主要用途是可以定义其基类中\_\_\_\_。

答案: (P127)不具有的成员

【解析】继承的特点，扩充基类，即在派生类中增加基类不具有的成员。

11. 在用class定义一个类时，数据成员和成员函数的默认访问权限是\_\_\_\_。

答案: (P69)private

【解析】定义类时的成员默认为私有，而结构体则是公有。

12. 使用new为int数组动态分配10个存储空间是\_\_\_\_。

答案: (P10)new int [10] ;

【解析】new delete动态开辟空间和删除空间。new int [10]，注意不要写成new int (10)，使用小括号只能开辟一个空间，使用10来初始化该值。

13. 类模板用来表达具有\_\_\_的模板类对象集。

答案: (P145)相同处理方法

【解析】模板特点是不同的数据具有相同的处理方法的抽象。

14. 如果通过同一个基类派生一系列的类，则将这些类总称为\_\_\_。

答案: (P174)类族

【解析】单继承方式派生的众多的类。

15. 面向对象的四个基本特性是多态性、继承性、和封装性\_\_\_。

答案: (P37)抽象

【解析】考察面向对象的四个特征。程序由一组抽象的对象组成，一组对象的共同特征抽象出类的概念，类是对象的抽象，对象是类的实例。封装即将数据和操作紧密结合提供访问的接口，外部通过接口实现访问数据，提供安全性。继承继承解决了类的扩展性。多态性不同对象调用相同的函数名，但调用不同的函数，实现不同的功能，解决了接口统一的问题。

16. 所有模板都是以\_\_\_关键字和一个形参表开头的。

答案: (P61)template

【解析】类模板，函数模板都要使用template这一关键字。

17. 在C++语言中，访问一个对象的成员所用的成员运算符是\_\_\_。

答案: .

【解析】成员运算符“.”，如果是指针可以使用“->”。

18. 开发一个C++语言程序的步骤通常包括编辑、\_\_\_、连接、运行和调试。

答案: (P21)编译

【解析】vc开发过程，要经过编辑、编译、连接和运行四个步骤，与其它高级语言相同。

19. 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。

20. 定义\_\_\_函数时，应在参数个数或参数类型上有所不同。

答案: (P59~60)重载

【解析】重载函数要求同名函数具有相同的功能，而只能是函数类型、参数个数或参数顺序不同。系统将根据同名函数的这些不同之处来选择其对应的实现。

三、改错题（本大题共5小题，每小题4分，共20分）

1. #include <iostream>

#include <fstream>

#include <string>

using namespace std;

class A

{public:

A(const char \*na){strcpy(name,na);}

private:

char name [80] ;

};

class B:public A

```

{ public:

B(const char *nm):A(nm){}

void show();

};

void B::show()

{ cout<<"name:"<<name<<endl;

}

void main()

{ B b1("B");

b1.show();

}

```

答案: private:因为name如果是私有的, 在派生类中无法访问, 而基类没有提供成员函数来访问name, 所以更改name访问权限为公有或保护, 这样对于派生类来说是透明的。

[修改] public: 或protected:

2. 下面的程序有错误, 请修改。

```

#include <iostream.h>

class A

{private:

int a;

public:

void func(B &);

A(int i){a=i;}

};

class B

{private:

int b;

friend void A::func(B &);

public:

B(int i){b=i;}

};

void A::func(B& r)

{a=r.b;

cout<<a<<endl;

}

void main()

{ B bt(3);

A at(10);

at.func(bt);

}

```

答案: [修改] class B;

class A

[解析] class A类A中使用B类中的成员增加对B声明。

3. #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;

4. #include <iostream.h>

class A

{private:

int x,y;

public:

void fun(int i,int j)

{x=i;y=j;}

void show()

{cout<<x<<" "<<y<<endl;}

};

void main()

{A a1;

a1.fun(2);

a1.show();

}

```

答案：void fun(int i,int j)调用时有一个参数，形参有两个，可以使第二个带默认值。

```

[修改] void fun(int i,int j = 0)

5. #include <iostream.h>

class A

{private:

int x;

protected:

int y;

public:

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

};

class B:public A

{public:

B(int a,int b):A(a,b){}

void show(){cout<<x<<,<<y<<endl;}

```

```
};

void main()

{B b(8,9);

b.show();

}
```

答案：private:在基类中是私有成员，即使采用公有派生，但在派生类无法访问。

【修改】 public:或protected:

四、完成程序题(本大题共5小题，每小题4分，共20分)

1. 完成下面类中成员函数的定义。

```
#include <iostream.h>

class vehicle

{protected:

int size;

int speed;

public:

void set(int s){speed=s;}

_____get(){return speed/10;}

};

class car:public vehicle

{ public:

int get(){return speed;}

};

class truck:public vehicle

{ public:

int get(){return speed/2;}

};

int max(_____)

{ if(v1.get()>v2.get())

return 1;

else

return 2;

}

void main()

{ truck t;

car c;

t.set(160);

c.set(80);

cout<<max(t,c)<<endl;//此结果输出为2

}
```

答案： virtual int, vehicle &v1,vehicle &v2

【解析】在基类和派生类都有get函数，输出结果是2，只有当这两个不同类型的对象，调用不同类的get函数，才能使结果为2，这就是多态性。所以将基类get定义为虚函数。max函数使用基类对象的引用的方法来实现。

2. 完成下面类中成员函数的定义。

```

#include <iostream>

#include <string>

using namespace std;

class str

{private:

char *st;

public:

str(char *a)

{set(a);

}

str & operator=(____)

{delete st;

set(a.st);

return *this;

}

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

~str(){delete st;}

void set(char *s)//初始化st

{____

strcpy(st,s);

}

};

void main()

{str s1("he"),s2("she");

s1.show(),s2.show();

s2=s1;

s1.show(),s2.show();}

```

答案：str &a, st=new char [strlen(s)+1] ；

〔解析〕对“=”运算符进行重载，调用时s2=s1,都是对象，所以形参使用对象的引用，不要使用对象作为形参（产生临时对象）。使用strcpy进行字符的复制，st必须有一定的空间，空间是strlen(s)+1（‘\0’作为结束符，strlen得到的长度不包括结束符）。

3. 下面程序段用来求三角形的面积，首先判断三边不符合组成三角形时，返回-1，符合时输出三角形面积。

```

#include <iostream.h>

#include <math.h>

double area(double a,double b,double c)

{if(____)

return -1;

else

{

double ar,l;

l=(a+b+c)/2;

ar=sqrt(l*(l-a)*(l-b)*(l-c));

return ar;
}
}

```



```

}

}

void main()

{double i=0,j=0,k=0;

cout<<"输入三角形三边: ";

cin>>i>>j>>k;

double s=area(i,j,k);

if(s<0)

cout<<"不是三角形"<<endl;

else

_____

}

```

答案:  $a+b>c||a+c>b||b+c>a$ , `cout<<s<<endl;`

【解析】三角形组成规则: 两边之和大于第三边。 $s<0$ 不是三角形, 是则输出s。

4. 在下面程序横线处填上适当内容, 使程序执行结果为:

```

S=2

S=5

S=9

#include <iostream.h>

void sum(int i)

{static int s;

_____

cout<<"s="<<s<<endl;

}

void main (void)

{int i;

for (i=0;_____)

sum(i);

}

```

答案:  $s=s+i+2;$ ,  $i<3,i++$

【解析】根据结果和调用形式, 得出规律。注意静态成员能保留上次运行的结果。循环了3次, 退出循环的条件。

5. 下面是一个三角形三边, 输出其面积C++程序, 在下划线处填上正确的语句。

```

#include <iostream.h>

#include <math.h>

void area()

{double a,b,c;

cout<<"Input a b c:";

_____

if(a+b>c&& a+c>b&& c+b>a)

{double l=(a+b+c)/2;

_____

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

}

```

```
else  
  
cout<<"Error"<<endl;  
  
}
```

```
void main()  
  
{area();}
```

答案：cin>>a>>b>>c;, double s=sqrt((l-a)\*(l-b)\*(l-c));

【解析】输入三个边的长度，由公式得出三角形的面积double s=sqrt((l-a)\*(l-b)\*(l-c));

## 五、程序分析题(本大题共2小题，每小题5分,共10分)

1. 给出下面程序输出结果。

```
#include <iostream.h>  
  
class example  
  
{int a;  
  
public:  
  
example(int b=5){a=b++;}  
  
void print(){a=a+1;cout <<a<<"";}  
  
void print()const  
  
{cout<<a<<endl;}  
  
};  
  
void main()  
  
{example x;  
  
const example y(2);  
  
x.print();  
  
y.print();  
  
}
```

答案：62

【解析】x是普通对象，调用普通的print函数；而y常对象，调用常成员函数。

2. 给出下面程序输出结果。

```
#include <iostream.h>  
  
class A  
  
{public:  
  
A()  
  
{cout<<"A 构造函数 \ n";fun();}  
  
virtual void fun()  
  
{cout<<"A::fun() 函数 \ n";}  
  
};  
  
class B:public A  
  
{public:  
  
B()  
  
{cout<<"B构造函数 \ n";fun();}  
  
void fun() {cout<<"B::fun() calle函数 \ n";}  
  
};  
  
void main()  
  
{B d;}
```

答案：A构造函数

A::fun()函数

B构造函数

B::fun()calle函数

【解析】定义派生类对象，首先调用基类构造函数，调用A类中fun()，然后调用B类的构造函数，在调用B的fun函数。

六、程序设计题（本大题共1小题，共10分）

1. 写一个程序，定义一个抽象类Shape,由它派生3个类：Square(正方形)、Trapezoid（梯形）和Triangle三角形。用虚函数分别计算几种图形面积、并求它们的和。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。

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

```
class Shape
```

```
{public:
```

```
virtual double area()const=0;
```

```
};
```

答案：class Square:public Shape

```
{public:
```

```
Square(double s):side(s){}
```

```
double area() const{return side*side;}
```

```
private:
```

```
double side;
```

```
};
```

```
class Trapezoid:public Shape
```

```
{public:
```

```
Trapezoid(double i,double j,double k):a(i),b(j),h(k)
```

```
{}
```

```
double area() const{return ((a+b)*h/2);}
```

```
private:
```

```
double a,b,h;
```

```
};
```

```
class Triangle:public Shape
```

```
{public:
```

```
Triangle(double i,double j):w(i),h(j)
```

```
{}
```

```
double area() const{return(w*h/2);}
```

```
private:
```

```
double w,h;
```

```
};
```

```
void main()
```

```
{Shape *p [5] ;
```

```
Square se(5);
```

```
Trapezoid td(2,5,4);
```

```
Triangle te(5,8);
```

```
p [0] =&se;
```

```
p [1] =&td;

p [2] =&te;

double da=0;

for(int i=0;i<3;i++)

{da+=p [i] ->area();}

cout<<"总面积是: "<<da<<endl;

}__
```