2009年全国C++程序设计模拟试卷（一）

一、单项选择题（本大题共20小题，每小题1分，共20分）在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

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项的表达式的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ClassY:publicX ，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)+jp ［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一个实体

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<<end1 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，可以调用___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的抽象，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)ector <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=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)

1.　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;
}__
一、单项选择题（本大题共 20小题，每小题 1分，共 20分) 在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。
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. 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。
3. 有关多态性说法不正确的是（）
A. C++语言的多态性分为编译时的多态性和运行时的多态性

B. 编译时的多态性可通过函数重载实现
C. 运行时的多态性可通过模板和虚函数实现
D. 实现运行时多态性的机制称为动态多态性
答案：C
解析：(P171)多态性分为静态的和动态的。静态通过函数的重载来实现，动态是通过基类指针或基类引用和虚函数来实现的。所以错误的是C项。
4. 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。
5. 类MyA的拷贝初始化构造函数是 （）
A. MyA()
B. MyA(MyA*)
C. MyA(MyA&)
D. MyA(MyA)
答案：C
解析：(P80) 复制即拷贝构造函数使用对象的引用作形参，防止临时产生一个对象， A无参构造函数，B是指针作为形参， D项是对象，所以选择C项。
6. 在C++中，函数原型不能标识（）
A. 函数的返回类型
B. 函数参数的个数
C. 函数参数类型
D. 函数的功能
答案：D
解析：函数的声明，说明函数的参数、返回类型以及函数名，函数体即实现部分决定功能。所以函数的原型不能决定函数的功能。
7. 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。
8. 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。

9. 在int a=3,int *p=&a ；中， *p的值是（）
A. 变量a的地址值
B. 无意义
C. 变量p的地址值
D. 3
答案：D
解析：*p代表引用a变量的值，p代表a的地址值。所以选择D项。
10. 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。
11. 如果从原有类定义新类可以实现的是（）
A. 信息隐藏
B. 数据封装
C. 继承机制
D. 数据抽象
答案：C
解析：(P129)继承指在原有类的基础上产生新类。数据封装即数据和操作组合在一起，形成类。信息的隐藏，通过访问权限来实现。数据抽象，将事物的特征抽象为数据成员或服务。因此选择C项。
12. 下面有关类说法不正确的是（）
A. 一个类可以有多个构造函数
B. 一个类只有一个析构函数
C. 析构函数需要指定参数
D. 在一个类中可以说明具有类类型的数据成员
答案：C
解析：(P80)构造函数可以有参数、可以重载、因此可以有多个， A项正确。析构函数只有一个不能重载、不能继承，没有返回值， B项正确，C项错误。
13. 在函数定义中的形参属于（）
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：形参或函数中定义的变量都是局部变量。在函数外定义的变量是全局变量。形参只能用局部变量，频繁使用的变量可以声明为寄存器变量，形参不能使用静态变量或寄存器变量。
14. 下列有关重载函数的说法中正确的是（）
A. 重载函数必须具有不同的返回值类型
B. 重载函数参数个数必须相同
C. 重载函数必须有不同的形参列表
D. 重载函数名可以不同
答案：C
解析：(P59)函数的重载必须函数名相同而形参类型或个数不同，与返回值无关。
15. this 指针存在的目的是（）

A. 保证基类私有成员在子类中可以被访问
B. 保证基类保护成员在子类中可以被访问
C. 保证每个对象拥有自己的数据成员，但共享处理这些数据成员的代码
D. 保证基类公有成员在子类中可以被访问
答案：C
解析：(P86)C++要求函数在被调用之前，应当让编译器知道该函数的原型，以便编译器利用函数原型提供的信息去检查调用的合法性，强制参数转换成为适当类型，保证参数的正确传递。对于标准库函数，其声明在头文件中，可以用 #include 宏命令包含这些原型文件；对于用户自定义函数，先定义、后调用的函数可以不用声明，但后定义、先调用的函数必须声明。一般为增加程序的可理解性，常将主函数放在程序开头，这样需要在主函数前对其所调用的函数一一进行声明，以消除函数所在位置的影响。所以选择 C 项。

16. 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。

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. 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; 。

19. C++类体系中，不能被派生类继承的有（）
A. 常成员函数
B. 构造函数
C. 虚函数
D. 静态成员函数
答案：B

解析：(P132)构造函数不能被继承。

20. 静态成员函数不能说明为（）

A. 整型函数

B. 浮点函数

C. 虚函数

D. 字符型函数

答案：C

解析：(P108)使用关键字static 声明的成员函数就是静态成员函数，静态成员函数也属于整个类而不属于类中的某个对象，它是该类的所有对象共享的成员函数。

静态成员函数可以在类体内定义，也可以在类外定义。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。虚函数是非静态的、非内联的成员函数。静态成员函数不能被说明为虚函数。

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. 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. 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. 一个抽象类的派生类可以实例化的必要条件是实现了所有的 ___。

答案：(P173)纯虚函数的定义

［解析］抽象类只因有纯虚函数，所以不能被实例化，所以派生类要实例化必须对纯虚函数进行定义。

4. 下面程序的输出结果为 ___。

```
#include <iostream.h>
void main()
{int num=2,i=6;
do
{i--;
num++;
}while(--i);
cout<<num<<endl;
}
```

答案：5

［解析］do—while 循环，前缀先使i 减少1后判断是否为零，不为零时再次执行循环，为零退出循环。循环值执行3次就退出，所以结果为5。

5. 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。

6. 在用C＋＋进行程序设计时，最好用 ___代替malloc 。

答案：(P10)new

［解析］new与delete 是C++语言特有的运算符，用于动态分配和释放内存。 new用于为各种数据类型分配内存，并把分配到的内存首地址赋给相应的指针。 new的功能类似于malloc （）函数。

使用new的格式为：
<指针变量>new<数据类型>;
其中，<数据类型>可以是基本数据类型，也可以是由基本类型派生出来的类型；<指针变量>取得分配到的内存首地址。new有3种使用形式。
（1）给单个对象申请分配内存
int *ip;ip=new int;//ip　　指向1个未初始化的int 型对象
（2）给单个对象申请分配内存的同时初始化该对象
int *ip;ip=new int(68);//ip　　指向1个表示为68的int 型对象
（3）同时给多个对象申请分配内存
int *ip;ip=new int　　［5］;//ip　指向5个未初始化的int 型对象的首地址
for(int i=0;i<5;i++)ip　　［i ］=5*i+1;//　给ip 指向的5个对象赋值
用new申请分配内存时，不一定能申请成功。若申请失败，则返回 NULL，即空指针。因此，在程序中可以通过判断new的返回值是否为0来获知系统中是否有足够的空间供用户使用。

7.　由const 修饰的对象称为 ___。
答案：(P113)常对象
［解析］使用const 关键字说明的成员函数称为常成员函数，使用 const 关键字说明的对象称为常对象。
常成员函数的说明格式如下：<返回类型说明符><成员函数名>(<参数表>)const;
常成员函数不更新对象的数据成员，也不能调用该类中没有用 const修饰的成员函数。常对象只能调用它的常成员函数，而不能调用其他成员函数。 const 关键字可以用于参与重载函数的区分。

8.　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。

9.　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

10.　不同对象可以调用相同名称的函数，但执行完全不同行为的现象称为　___。
答案：(P167)多态性
［解析］多态性的概念。虚函数是实现多态的基础，运行过程中的多态需要同时满足 3个条件：(1) 类之间应满足子类型关系。(2) 必须要有声明的虚函数。(3) 调用虚函数操作的是指向对象的指针或者对象引用；或者是由成员函数调用虚函数（如果是在构造函数或析构函数中调用虚函数，则采用静态联编）。

11.　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。

12.　预处理命令以 ___符号开头。
答案：(P183)operater
［解析］文件包含、预处理和编译都是以＃开头。

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.　C++程序的源文件扩展名为 ___。
答案：(P21)cpp
［解析］源程序*.cpp，目标文件为*.obj，可执行程序*.exe 。

15.　在#include 命令中所包含的头文件，可以是系统定义的头文件，也可以是　___的头文件。
答案：(P7) 自定义
［解析］＃include 装入文件有两种方式<>和“ ”，一是系统的，一是自定义文件。

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. 语句序列

ifstream infile;

infile.open(“data.dat ”);

的功能可用一个语句实现，这个语句是 ___。

答案：(P199)ifstream infile(“data.dat ”);

［解析］void ifstream::open(const char *fname,int mode=ios::in,int access=filebuf::openprot);

ifstream::ifstream(const char *fname,int mode=ios::in,int access=filebuf::openprot);

其中，第一个参数是用来传递文件名的；第二个参数 mode的值决定文件将如何被打开；第三个参数access的值决定文件的访问方式，一般取缺省值 filebuf::openprot ，表示是普通文件。

mode的取值如下：(1)ios::in ：打开一个文件进行读操作，而且该文件必须已经存在；(2)ios::nocreate ：不建立新的文件。当文件不存在时，导致 open()失败；(3)ios::noreplace ：不修改原来已经存在的文件。若文件已经存在，导致 open()失败；(4)ios::binary ：文件以二进制方式打开，缺省时为文本文件。

19. 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：：。

20. 在编译指令中，宏定义使用 ___指令。

答案：(P6、97)#define

［解析］静态成员是所有对象共享的特征，也就是类的特征。

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>

```
#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. #include <iostream.h>
void f(int *a,int n)
{int i=0,j=0;
int k=0;
for(;i<n/2;i++)
{k=a［i］;
a［i］=a［n-i-1 ］;
a［n-i-1 ］=k;
}
}
void show(int a ［］,int n)
{for(int i=0;i<n;i++)
cout<<a［i］<<" ";
cout<<endl;
}
void main()
{int p ［5］;
int i=0,n=5;
for(;i<5;i++)
{p ［i］=i;}
f(*p,n);
show(p,n);
答案： ［修改］f(p,n);
［解析］f(*p,n);f 函数第一个参数是指针而调用时使用*p，*p表示p所指向的变量或对象，不是地址即不是指针。

3. #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;

4. #include <iostream.h>
void main()
{int i,*p;
i=10;
*p=i;
cout<<*p<<endl;
}
答案：*p=i; 指针即地址没有被赋值。
［修改］p=&i;

5. #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)

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>
#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）。

2. 一个类的头文件如下所示，num初始化值为 5，程序产生对象 T，且修改 num为10，并使用show()函数输出num的值10。

```
#include <iostream.h>
class Test
{private:
static int num;
public:
Test(int);
void show();
};
_____
Test::Test(int n)
{num=n;}
void Test::show()
{cout<<num<<endl;}
void main()
{Test t(10);
_____
}
```

答案：int Test::num=5;　，t.show();

［解析］静态成员在类外初始化，注意它的格式。调用 show输出。

3.　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*(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*(l-a)*(l-b)*(l-c));

4.　下面程序中 Base是抽象类。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:

```
Der1 called!
Der2 called!
#include <iostream.h>
class Base
{public:
_____;
};
class Der1:public Base
{public:
void display(){cout<<"Der1 called!"<<endl;}
```

```
};
class Der2:public Base
{public:
void display(){cout<<"Der2 called!"<<endl;}
};
void fun(_____________)
{p->display();}
void main()
{Der1 b1;
Der2 b2;
Base * p=&b1;
fun(p);
p=&b2;
fun(p);
}
```

答案：virtual void display()=0;　　，Base *p

［解析］抽象类有纯虚函数，派生类为 display 。结果fun函数用指针做参数。

5.　下面程序中用来求数组和。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:s ＝150。

```
#include <iostream.h>
class Arr
{int *a,n;
public:
Arr():a(0),n(0){}
Arr(int *aa, int nn)
{n=nn;
a=new int ［n］;
for(int i=0;i<nn;i++)
*(a+i)=*(aa+i);
}
~Arr(){delete a;}
_____________;
{return *(a+i);}
};
void main()
{int b　［5］={10,20,30,40,50};
Arr a1(b,5);
int i=0,s=0;
_____________
s+=a1.GetValue(i);
cout<<"s="<<s<<endl;
}
```

答案：int GetValue(int i)　　，for(;i<5;i++)

［解析］函数调用GetValue,由此可知要定义该函数，循环求和，循环 5次。

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 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>
void main()
{ int *p1;
int **p2=&p1;
int b=20;
p1=&b;
cout<<**p2<<endl;
}
```

答案：20

［解析］p1指向b，而p指向p1的地址。*p2表示p1的地址，p1的地址就是&b，即*p2是&b,所以**p2就是b变量的值。

3. 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。

4. 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. 编写类 String 的构造函数、析构函数和赋值函数和测试程序。
已知类String 的原型为：

```
#include <iostream.h>
#include <string.h>
class String
{public:
String(const char *str=NULL); //          普通构造函数
String(const String &other); //          拷贝构造函数
~String(); //       析构函数
String & operator=(const String &other); //             赋值函数
void show()
{cout<<m_data<<endl;
}
private:
char *m_data; //     用于保存字符串
```

```
};
答案：String::~String()
{delete ［］m_data;// 由于m_data是内部数据类型，也可以写成 delete m_data;
}
String::String(const char *str)
{if(str==NULL)
{m_data=new char［1］;// 若能加NULL判断则更好
*m_data=＼0;
}
else
{int length=strlen(str);
m_data=new char［length+1］; // 若能加NULL判断则更好
strcpy(m_data, str);
}
}
String::String(const String &other)
{int length=strlen(other.m_data);
m_data=new char［length+1］;// 若能加NULL判断则更好
strcpy(m_data, other.m_data);
}
String & String::operator=(const String &other)
{if(this==&other)
return *this;
delete ［］m_data;
int length=strlen(other.m_data);
m_data=new char［length+1］;// 若能加NULL判断则更好
strcpy(m_data, other.m_data);
return *this;
}
void main()
{String str1("aa"),str2;
str1.show();
str2=str1;
str2.show();
String str3(str2);
str3.show();
}__
```

2010年全国 C++程序设计模拟试卷（三)

一、单项选择题（本大题共 20小题，每小题 1分，共 20分) 在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

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，形参个数为 ___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;
}__
```

2009年全国 C++程序设计模拟试卷（四）

一、单项选择题（本大题共 20小题，每小题 1分，共 20分）在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

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. 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. 函数调用 func((exp1,exp2),(exp3,exp4,exp5)) 中所含实参的个数为（）

A. 1

B. 2

C. 4

D. 5

答案：B

解析：(exp1,exp2)、(exp3,exp4,exp5) 表示是两个逗号表达式，值是最后一个值，相当于两个参数。因此实参的个数是2。

4. 设有函数模板

```
template <class Q>
Q Sum(Q x,Q y)
{return (x)+(y);}
```

则下列语句中对该函数模板错误的使用是（）

A. Sum(10,2);

B. Sum(5.0,6.7) ;

C. Sum(15.2f,16.0f);

D. Sum(“AB”,”CD”);

答案：D

解析：(P40)由Q Sum(Q x,Q y)可知形参和函数返回值都是同一种数据类型。 A、 B、 C三项都正确。而D项用字符串作为实参，字符串的操作与数值类型不同，要用特殊方法进行字符串的连接和运算。

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所指的对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. 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。

8. 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。

9. 有关C++编译指令，以下叙述正确的是（）

A. C++每行可以写多条编译指令

B. #include 指令中的文件名可含有路径信息

C. C++的编译指令可以以 #或// 开始

D. C++中不管＃ if 后的常量表达式是否为真，该部分都需要编译

答案：B

解析：(P96)编译指令以＃作为开头，只能一行写一条， ＃ if 有选择进行编译，所以选择B项。

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. 下列关于静态数据成员的描述中正确的是（）
A. 静态数据成员是类的所有对象所共有的
B. 静态数据成员要在构造函数内初始化
C. 类的每个对象有自己的静态数据成员
D. 静态数据成员不能通过类的对象调用
答案：D
解析：(P107)静态成员属于类的即所有对象所共享的，只能在外部进行初始化。使用时可以使用形式有两种，类名:: 静态成员或者对象. 静态成员。所以选择D项。
13. 在编译指令中，宏定义使用哪个指令（）
A. #if
B. #include
C. #define
D. #error
答案：C
解析：(P7)#if 条件编译，＃include 文件包含，＃error 错误处理。
14. 类的析构函数是对一个对象进行以下哪种操作时自动调用的是（）
A. 建立
B. 撤销
C. 赋值
D. 引用
答案：B
解析：(P80)删除对象或结束程序时，自动调用析构函数。
15. 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：(P107)静态成员的特性是静态成员只有一个拷贝（副本），这个副本被所有属于这个类的对象共享。这种共享与全局变量或全局函数相比，既没有破坏数据隐藏的原则，又保证了安全性。静态成员表示整个类范围的信息，其声明以关键字 static 开始，包括静态数据成员和静态成员函数。在对静态数据成员初始化时应注意：
(1) 应在类体外对静态数据成员进行初始化（静态数据成员的初始化与它的访问控制权限无关）。
(2) 静态数据成员初始化时前面不加 static 关键字，以免与一般静态变量或对象混淆。
(3) 由于静态数据成员是类的成员，因此在初始化时必须使用作用域运算符 (::) 限定它所属的类。因此选择D项。

16. 如果类A被说明成类 B的友元，则（）
A. 类B不一定是类 A的友元
B. 类B的成员即类 A的成员
C. 类A的成员即类 B的成员
D. 类A的成员函数不得访问类 B的成员
答案：A
解析：(P113)友元关系不能被继承，友元关系是单向的，友元关系不具有传递性。但是友元函数不是类的成员，所以选择A项。
17. 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。
18. 下列运算符中，在 C++语言中不能重载的是（）
A. *
B. >=
C. ：：
D. /
答案：C
解析：(P186)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。
19. 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。
20. 在C++语言中，数据封装要解决的问题是（）
A. 数据的规范化
B. 便于数据转换
C. 避免数据丢失
D. 防止不同模块之间数据的非法访问
答案：D
解析：(P39)封装是指把对象属性和操作结合在一起，构成独立的单元，它的内部信息对外界是隐蔽的，不允许外界直接存取对象的属性，只能通过有限的接口与对象发生联系。类是数据封装的工具，对象是封装的实现。类的访问控制机制体现在类的成员中可以有公有成员、私有成员和保护成员。对于外界而言，只需要知道对象所表现的外部行为，而不必了解内部实现细节。封装体现了面向对象方法的 “ 信息隐蔽和局部化原则 ”。

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. 若要使用 string 类，需要引入的头文件是 ___。
答案：(P40)string.h
［解析］编译时要将头文件引入才能使用标准库中的方法或成员。

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. 向量操作方法中 ___方法返回向量中的第一个对象。
答案：(P151)front
［解析］考察向量方法的使用。 front(): 返回向量中的第1个对象。back（）：返回向量中的最后一个对象。operator ［］(size_type,n): 返回向量中的第n＋1个对象（下标为n的向量元素）。
4. C++派生类使用两种基本的面向对象技术：第一种称为性质约束，即对基类的性质加以限制；第二种称为___，即增加派生类的性质。
答案：(P129)性质扩展
［解析］派生类通过继承可以从基类中获得成员，也可以自定义成员。
5. 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。
6. 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。
7. 预处理语句有三种，分别是宏定义、文件包含和 ___。
答案：(P7)条件编译
［解析］宏定义＃define，文件包含＃include 和条件编译#if 等。
8. 构造函数、析构函数和友元函数中，不是该类成员的是 ___。
答案：(P109)友元函数
［解析］友元函数不是类成员，但可以访问类中成员。
9. 控制格式输入输出的操作中，函数 ___是设置域宽的。要求给出函数名和参数类型）。
答案：(P193)setw(int)
［解析］setw(int n): 用来设置n输出宽度。
10. 派生类的成员一般分为两部分，一部分是 ___，另一部分是自己定义的新成员。
答案：(P127)从基类继承的成员
［解析］派生类成员一个来自继承基类成员，一个来自本身增加的成员。
11. C＋＋中ostream的直接基类 ___。
答案：(P193)ios
［解析］istream 和ostream的直接基类是ios 。
12. vector 的___方法返回向量中的最后一个对象。
答案：(P151)back
［解析］front(): 返回向量中的第1个对象。back（）：返回向量中的最后一个对象。operator ［］(size_type,n): 返回向量中的第n＋1个对象（下标为n的向量元素）。
13. 执行下列代码

```
int i=230;
cout <<"i="<<hex <<i<<endl;
```

程序的输出结果为___。
答案：(P193)i=e6
［解析］流类库中使用格式符，输出十六进制数据。

14. 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。
15. 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。
16. 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。
17. 定义虚函数所用的关键字是 ___。
答案：(P170)virtual
［解析］在成员函数前加virtual 修饰的函数就是虚函数。但不是所有成员函数都可以定义为虚函数的。比如构造函数，不能定义虚函数。
18. 执行下列代码
cout<<“ oct: ”<<oct<<34;
程序的输出结果是___。
答案：(P193)Oct:42
［解析］oct 表示八进制，hex表示十六进制，但它们只能输出整型的数据。
19. 在C++中要创建一个文件输入流对象 fin ，同时该对象打开文件“ Test.txt ”用于输入，则正确的声明语句是___。
答案：(P200)ifstream fin(“ Test.txt ”)；
［解析］文件操作中ifstream 用于文件的输入，可以调用它的构造函数与要打开的文件进行关联
20. 如果一个派生类只有一个唯一的基类，则这样的继承关系称为 ___。
答案：(P130)单一
［解析］根据派生类所拥有的基类数目不同，可以分为单继承和多继承。一个类只有一个直接基类时，称为单继承；而一个类同时有多个直接基类时，则称为多继承。
基类与派生类之间的关系如下：
(1) 基类是对派生类的抽象，派生类是对基类的具体化，是基类定义的延续。
(2) 派生类是基类的组合。多继承可以看作是多个单继承的简单组合。
(3) 公有派生类的对象可以作为基类的对象处理。
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. 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 T
{protected:
int p;
public:
T(int m){p=m;}

```
};
void main()
{ T a(10);
cout<<a.p<<endl;
}
```

答案：［修改］public

［解析］protected 保护类型的成员，不能在类外访问。

3.
```
#include <iostream>
using namespace std;
class Date;
class Time
{public:
Time(int h,int m,int s)
{hour=h,minute=m,sec=s;}
void show(Date & d);
private:
int hour,minute,sec;
};
class Date
{public:
Date(int m,int d,int y)
{month=m,day=d,year=y;}
void Time::show(Date &);
private:
int month,day,year;
};
void Time::show(Date & d)
{cout<<d.month <<"-"<<d.day<<"-"<<d.year<<endl;
cout<<hour<<":"<<minute<<":"<<sec<<endl;
}
void main()
{Time t1(9,23,50);
Date d1(12,20,2008);
t1.show(d1);
```

答案：void Time::show(Date &); 成员函数作为友元函数，要加 friend 。

［修改］friend void Time::show(Date &);

4. 输出最小值，有一处错误。

```
#include <iostream.h>
class Test
{int a,b;
int getmin()
{return (a<b?a:b);}
public:
int c;
void setValue(int x1,int x2,int x3)
{a=x1;b=x2;c=x3;}
int GetMin();
};
int Test::GetMin()
{int d=getmin();
```

```
return (d=d<c?d:c);
}
void main()
{Test t1;
t1.setValue(34,6,2);
cout<<t1.getmin ()<<endl;
}
```

答案：cout<<t1.getmin()<<endl;　采用默认的访问权限即私有的，在外部无法访问。

［修改］cout<<t1.GetMin()<<endl;

5.　实现数值、字符串的交换。

```
#include <iostream>
#include <string>
using namespace std;
template<class T>
void Swap(T& a,T& b)
{T temp;
   temp=a,a=b,b=temp;
}
void main()
{int a=5,b=9;
char s1 []="Hello",s2 []="hi";
Swap(a,b);
Swap(s1,s2);
cout<<"a="<<a<<",b="<<b<<endl;
cout<<"s1="<<s1<<",s2="<<s2<<endl;
}
```

答案：char s1 []="Hello",s2 []="hi"; 使用Swap(s1,s2)调用交换的是地址。字符指针作实参，形参值发生改变，实参也就发生变化。

［修改］char *s1="Hello",*s2="hi";

四、完成程序题（本大题共 5小题，每小题 4分，共 20分）

1.　在下划线处填上缺少的部分。

```
#include <iostream.h>
class A
{int a,b;
public:
_____; // 定义构造函数，使参数i 和j 的默认值为0
{a=i;b=j;}//     在函数体中用i 初始化a，用j 初始化b
};
main()
{A *p;
_____;// 调用带参构造函数生成由p指向的动态对象
// 使a和b成员分别被初始化为4和5
}
```

答案：A(int i=0,int j=0)　　，p=new A(4,5)

［解析］构造函数带默认参数为0，使用new运算符动态分配对象空间，同时初始对象成员值4，5。

2.　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。

3.　下面程序运行的结果是：　5+10=15。

```
#include <iostream.h>
class Test
{ private:
int x,y;
public:
Test() {x=y=0;}
void Setxy(int x,int y) {_______}
void show(){_______}
};
void main()
{Test ptr;
ptr.Setxy(5,10);
ptr.show();
}
```

答案：(*this).x=x; (*this).y=y;　　，cout<<x<<"+"<<y<<"="<<x+y<<endl;

［解析］形参同数据成员同名，使用 this 来访问。

4.　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。

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 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:
a(int i=10){x=i;cout<<"a:"<<x<<endl;}
int x;
};
class b:public a
{public:
b(int i):A(i){x=i;cout<<"b:"<<x<<", "<<a::x<<endl;}
private:
a A;
int x;
};
void main()
{b B(5);
}
```

答案：a:10
a:5
b:5,10

［解析］定义对象B，先调用基类构造函数，在b构造函数中使用的是A(i) ， 注意大小写，不是a(i)，也就是说调用基类的构造函数时没有实参值，所以采用默认值；在初始化类成员 A，即A(i) ， i ＝5，所以输入为a:5 ；最后是b类的构造函数， x=5,来自基类x＝10， 输出b:5,10 。

2. 运行程序，写出程序执行的结果。

```
#include<iostream.h>
class Location
{public:
int X,Y;
void init(int initX,int initY);
int GetX();
int GetY();
};
void Location::init (int initX,int initY)
{X=initX;
Y=initY;
}
int Location::GetX()
{return X;
}
int Location::GetY()
{return Y;
}
void display(Location& rL)
{cout<<rL.GetX()<<" "<<rL.GetY()<<    ＼n;
}
void main()
{Location A  ［5］={{5,5},{3,3},{1,1},{2,2},{4,4}};
```

```
Location *rA=A;
A［3］.init(7,3);
rA->init(7,8);
for (int i=0;i<5;i++)
display(*(rA++));
}
```

答案：7 8
3 3
1 1
7 3
4 4
［解析］对象数组的使用。使用数组对象修改了A［3］元素的值，又使用指针修改指针所指向的第一个元素的值，因此修改了A［0］和A［3］元素的值。

3. 给出下面程序输出结果。

```
#include <iostream.h>
int a ［8］={1,2,3,4,5,6,7};
void fun(int *pa,int n);
void main()
{int m=8;
fun(a,m);
cout<<a［7］<<endl;
}
void fun(int *pa,int n)
{for (int i=0;i<n-1;i++)
*(pa+7)+=*(pa+i);
}
```

答案：28
［解析］数组名与指针都表示地址，只是数组名是常地址，不能改变；指针是地址变量，使用时可以当数组名使用。

4. 给出下面程序输出结果。

```
#include <iostream.h>
class A
{int *a;
public:
A(int x=0):a(new int(x)){}
~A() {delete a;}
int getA() {return *a;}
void setA(int x) {*a=x;}
};
void main()
{A x1,x2(3);
A *p=&x2;
(*p).setA(x2.getA()+5);
x1.setA(10+x1.getA());
cout<<x1.getA()<<""<<x2.getA()<<endl;
}
```

答案：108
［解析］p指向对象x2，x2.getA()+5 该值为8 即x2.a＝8;10+x1.getA() 为10，x1.a＝10。

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;
}
```

2010年全国 C++程序设计模拟试卷（五)

一、单项选择题（本大题共 20小题，每小题 1分，共 20分) 在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

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，等价为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）请在每小题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)operater
［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)ector <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. 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　："<<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　。

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.."<<i<<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)();
}
```

答案：ft::y=3

［解析］指向函数的指针的使用， p指向fun1函数，（ft1.*p ）实际就是调用ft1 对象的fun1（）函数。

六、程序设计题（本大题共 1小题，共 10分）

1. 求n(n=3) 个学生的最高分和最低分及姓名，已有 student 类声明和 main函数，完成student 类的实现部分。

```
#include <iostream.h>
#include <string.h>
class student
{char name［10］;
int deg;
public:
student(char na ［］="",int d=0);
char * getname();
friend int compare(student &s1,student &s2);
int getdeg();
};
void main()
{student st ［］={student(" 王强",74),student(" 李刚",68),student(" 张雪",84)};
int i=0,min=0,max=0;
for(i=1;i<3;i++)
{if(compare(st ［max］,st ［i ］)==-1)
max=i;
if(compare(st ［min］,st ［i ］)==1)
min=i;
}
cout<<"最高分："<<st ［max］.getdeg()<<" 姓名:"<<st ［max］.getname()<<endl;
cout<<"最低分:"<<(*(st+min)).getdeg()<<" 姓名:"<<st ［max］.getname()<<endl;
```

}
答案：student::student(char na　　[] ,int d)
{strcpy(name,na);
deg=d;
}
char * student::getname(){return name;}
int compare(student &s1,student &s2)
{if(s1.deg>s2.deg)
return 1;
else if(s1.deg==s2.deg)

return 0;
else return -1;
}
int student::getdeg()
{return deg;}__

2010年全国 C++程序设计模拟试卷（六)

一、单项选择题（本大题共 20小题，每小题 1分，共 20分) 在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

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所指的对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一个实体
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。
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*(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*(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;
}__
```