一、            单选题，请选出最优的一个答案(1x10 = 10分)

1.  下列使用引用的方法**不正确**的是( )   
a) int a(0); int &ra=a; b) int a(0); int &ra(a); c) int a(0); int &ra; ra=a; d) int a(10);void f(int &ra){…}; f(a);

2. CMyClass是一个类，CMyClass \*p=null;;下面分配10个CMyClass对象所需内存的是( )：

a) p=new CMyClass[10]; b) p=new CMyClass(10); c) p=malloc CMyClass[10]; d) p=new sizeof(CMyClass)\*10;

3. 基类的析构函数一般应该设计为( ) a)私有的 b)受保护的 c)友元函数 d)虚函数

4. class X{public:static int count(); int f(); int x;};X obj; 下列写法**不正确**的是：

a)int c=obj.count(); b) int X::f(){return count();} c) int X::count(){return f();} d) int c=X::count();

5. int f(const int t) **不能**和下列哪个函数构成重载关系？( )

a) int f(int t,int w) ; b) const int f(x) ; c) int f(const int k,char c=’c’) ; d) int f(int t) ;

6. 下列哪个运算符不能通过成员函数重载( ) a)赋值运算符 b)输入运算符 c)移位运算符 d) 大于运算符

7. class CAR{public:B(int x){} ; } ; class TIRE{public:TIRE(int r); }; class BMW :public CAR{ int & ra;TIRE t; string owner;} ; 在BMW中a) 调用B(int x) b) 初始化ra c)初始化 t d)初始化owner 可以不用初始化列表。

8. this指针在访问属性上相当于( )成员 a)public b) friend c)protected d) e) static

9. typedef void (\*pf)(int x,int y);中pf是( ) a) 函数指针 b) 函数 c) 数据类型 d) 变量

10. 下列说法正确的是( ) a)派生类对象可给基类对象赋值 b) 基类指针可以给派生类指针赋值 c) 基类引用可以给派生类引用赋值 d) 基类对象一定属于派生类对象

二、            填空题，每空一分(1x10 = 10分)

1.      C++面向对象程序设计中，封装是通过\_\_\_\_\_\_\_实现的。

2.      inline函数的好处是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.      class X的拷贝构造函数的形参类形是\_\_\_\_\_\_\_\_\_\_\_或\_\_\_\_\_\_\_\_\_\_\_\_\_

4.      函数模版 \_\_\_\_\_\_\_\_\_\_\_\_\_\_void swap(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_){\_\_\_\_\_ c; c=a;a=b;b=c;}可以交换任意两个支持赋值运算的变量/对象。

5.      受保护派生类的成员函数可以直接访问基类的\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 成员变量/函数。

6.      class X{}; sizeof(X)=\_\_\_\_\_\_\_\_\_\_; class Y{virtual ~Y(){};}; sizeof(Y)=\_\_\_\_\_\_\_\_\_\_\_.

三、            改错题, **说出错误所在，并改正，答案写在右边** (10x2 = 20分)

1， 类的声明

class WRONG{

virtual void WRONG(int x);

virtual WRONG(WRONG other);

void ~WONG(int type);

public:

auto int x=5, y(0);

int getCount() const;

}

2， 继承与派生

class HOTEL{public:

HOTEL(){};\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

virtual BookRoom()=0;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int price;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

};

**class BUS**

{

string producer;

int price;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public:

BUS(int emission){}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

};

class BUSHOTEL:public HOTEL,public BUS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

{public:

BUSHOTEL(int emission ){};\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getPrice(){return HOTEL::price+BUS::price;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

};

BUSHOTEL romanticHotel;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

romanticHotel. producer=”ParisAuto”;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

四、            程序阅读题，读程序,**输出写在右边** (10+5+5= 20分)

**1．对象的构造、析构、静态对象、与全局对象**

#include "stdafx.h"

#include <IOSTREAM>

#include <STDLIB.H>

using namespace std;

class ANIMAL{public:

ANIMAL(){cout<<" Begin Animal "<<endl;}

virtual ~ANIMAL(){cout<<"End Animal"<<endl;}

};

class FISH:public ANIMAL{

public:

FISH(){cout<<"Free fish is born"<<endl;}

~FISH(){cout<<"Free fish is dead"<<endl;}

};

class EARTH{public:

EARTH(){cout<<"No earth,no life"<<endl;

FISH littleBlackFish;

}

~EARTH(){printf("2012 is coming\n");}

};

class GOD{

public:

GOD(){cout<<"Create world"<<endl;}

~GOD(){printf("Eternal GOD\n");}

void CreateWorld(){static EARTH planet; }

};

GOD god;

int main(int argc, char\* argv[])

{

printf("1st line of main\n");

god.CreateWorld();

printf("main exits\n");

return 0;

}

**2. 异常处理**

char accessMem(char \* p){

char m=\*p; return m;

}

int main() {

cout<<"Hello\n";

try{

cout<<"in try block\n";

char x='S';

cout<<accessMem(&x)<<endl;

char \* pNull=NULL;

cout<<accessMem(pNull)<<endl;

cout<<"end try block";

}

catch(char c){cout<<"Catch an exception : "<<c<<endl;}

catch (int i){cout<<"Catch an exception : "<<i<<endl; }

catch (...) {cout<<"unknown exception catched"<<endl; }

cout<<"Bye-bye\n";

return 0;

}

**3．继承与多态， 输出写在右边**

#include "stdafx.h"

#include <iostream>

#include <STRING>

using namespace std;

class ANIMAL{public:

virtual void echo(){cout<<"O-O"<<endl;}

~ANIMAL(){};

};

class DOG:public ANIMAL{

void echo(){cout<<"Wang-Wang"<<endl;} };

class DUCK:public ANIMAL{

void echo(){cout<<"Quack-Quack"<<endl;} };

class CAT:public ANIMAL{

void echo(){cout<<"Meeao-Meeao "<<endl;} };

void fobj(ANIMAL obj){obj.echo();}

void fref(ANIMAL & ref){ref.echo();}

void fpoint(ANIMAL \* p){p->echo();}

int main(){

cout<<"Animal music begin"<<endl;

DOG dog;DUCK duck;CAT cat;

fobj(cat);fobj(dog);fobj(duck);

fref(cat);fref(dog);fref(duck);

fpoint(&cat);fpoint(&dog);fpoint(&duck);

cout<<"Animal music begin"<<endl;

return 0;

}

五、            简答题 (5x4 = 20分)

1， 简单说明拷贝构造函数与赋值运算符完成的任务有何异同？

2， 函数重载和类继承中的函数重定义有何异同？

3， 虚函数、虚基类和抽象类有何区别和联系？

4， 静态多态和动态多态的概念，区别和联系？

六、            写程序 (5+15 = 20分)

1， 通过输入输出流的格式化输出，实现程序运行环境所能表示的任意一个浮点数的右对齐保留4位小数点输出。写出完整程序，输出数值由键盘输入。

2， 原点到平面任一点(x,y)的向量可以用VECTOR(x,y)来表示，试将点POINT声明为一个类，写一个向量类VECTOR2D，向量类可以用x，y坐标，POINT对象,或者另一个向量初始化，并完成两个向量相加“+”运算符的重载。不要求写完整的程序。

https://clients2.google.com/service/update2/crx?response=redirect&x=id%3D ohmmkhmmmpcnpikjeljgnaoabkaalbgc?hl=zh-CN%26uc