一、**（20分，除特殊说明外每小题2分）**

1、cout、endl等前加std::，或者使用iostream.h等老的头文件。

2、不能，派生类不能直接访问基类私有成员；

3、返回x的引用，后面函数调用表达式可做左值。或者说，可以绕过编译器的访问权限检查。

4、虚继承，可去掉，本题没有菱形继承带来的二义性。

5、声明常量字符串，即指针是变量，但其指向字符串是常量，不能去掉；

6、防止new不成功而带来的使用野指针错误；

7、成员函数增加this指针做参数，根据this找到某对象的数据成员；

8、先调用基类A构造函数，再调用派生类B构造函数，各两次。

9、2,WANG,0

3,LIU,4

**二、（42分，除特殊说明外每小题2分）**

1、将类型const char\*定义为charp，以方便使用；

2、为字符串结尾的空字符留位置。

3、删除name指向存储空间，防止内存泄漏。

4、常量成员函数，不允许修改数据成员。或答为，使this指针指向常量对象。

5、显式调用嵌套对象成员的构造函数，因为其需要传参数；

6、虚函数；

7、公有派生是运行时多态必要条件；

8、,author(au)；

9、重写；

10、重新定义。

11、基类对象的指针数组；

12、防止使用野指针；

13、若空，则为野指针，不能使用；

14、函数重载，编译器在编译时根据实参绑定。

15、for语句内；

16、C++,WANG,92,

Java,72,

destruct HandBook

destructor Name

destruct Book

destructor Name

destruct Book

destructor Name

destruct Library

17、

**图书馆对象lb**在stack（main的栈帧），数据成员都在其内部；

**lb**的数据成员s指向的数组在heap，而数组每个元素（指针）指向的**Book、HandBook对象**在heap。

Book、HandBook对象中有数据成员是Name对象，当然也在Book、HandBook对象内部，但Name对象中只有数据成员char\*，其指向的字符串在heap，不在该对象内部。

创建对象传入的实参"C++"、"WANG"在静态区。

所有成员函数在**代码区**。

**三、（15分，除特殊说明外每小题2分）**

1、函数模板；

2、显式实例化模板，不写会隐式实例化，对结果没影响，但**影响效率**;

3、形参是引用，不仅减少了拷贝构造带来的降低效率问题，更重要的是实现运行时多态；

4、0；

5、3；

6、x=1,y=A；

7、继承、组合（或者对象嵌套）、模板

**四、（12分）**

构造函数、拷贝构造函数、析构函数、<<重载等省略。另外，答案不唯一。

String& operator+(const String &a){

char\*p=new char[strlen(array)+strlen(a.array)+1];

strcpy(p,array);

strcat(p,a.array);

if (array)delete [] array;

array = p;

return \*this;

}

String operator+(char\* c){

String tmp(\*this);

char\* p=new char[strlen(tmp.array)+strlen(c)+1];

strcpy(p,tmp.array);

strcat(p,c);

delete [] tmp.array;

tmp.array=p;

return tmp;

}

**五、（11分）**

class Stu;

class Tea{

Stu\* stu;

public:

Tea(int n);

void add(int,Stu);

void display();

~Tea();

};

class Stu {

int score;

public:

Stu(int i=0){score=i;}

friend void Tea::display();

};

Tea::Tea(int n){stu=new Stu[n];}

void Tea::add(int i,Stu& s){stu[i]=s;}

void Tea::display(){

for (int i=0;i<2;i++) cout<<stu[i].score<<",";

cout<<endl;

}

Tea::~Tea(){delete [] stu;}