```cpp
#include <iostream>
using namespace std;
#include <vector>
#include<string>

class Item {
private:
    string title;
    int playingTime;
    bool gotIt = false;
    string comment;
public:
    Item(string title,int playingTime,bool gotIt,string comment) {
        this->title = title;
        this->playingTime = playingTime;
        this->gotIt = gotIt;
        this->comment = comment;
    }
    void print() {
        cout << "Item" << endl;
    };
};

class CD :public Item{
private:
    string  artist;
    int numofTracks;

public:
    CD(string title, string artist, int numofTracks, int playingTime, string
comment);

    //void print() {
    //  cout << title + ":" + artist<< endl;
    //}
};

CD::CD(string title, string artist, int numofTracks, int playingTime, string
comment) :Item(title,playingTime,false,comment) {
    {
        this->artist = artist;
        this->numofTracks = numofTracks;
    }
}
class DVD :public Item{
private:
    string title, director, comment;
    int  playingTime;
    bool gotIt = false;
public:
    DVD(string title, string director, int playingTime, string comment);
};
```

```
DVD::DVD(string title, string director, int playingTime, string comment){
    this->title = title;
    this->playingTime = playingTime;
    this->comment = comment;
    this->director = director;
    }
    //void print() {
    //  cout << title + ":" + director << endl;
    //}
class Database {
private:
    vector<Item> listItem;
public:
    /*void add(CD cd) {
        listCD.push_back(cd);
    }
    void add(DVD dvd) {
        listDVD.push_back(dvd);
    }*/
    void add(Item item) {
        listItem.push_back(item);
    }
    void list() {
        /*for (CD cd : listCD) {
            cd.print();
        }
        for (DVD dvd : listDVD) {
            dvd.print();
        }*/
        for (Item item : listItem) {
            item.print();
        }
    }
};



int main() {
    Database db;
    CD cd("title", "artist", 4, 60, "comment");
    cd.print();
    DVD dvd("xxx", "aaa", 60, "comment");
    db.add(cd);
    db.add(dvd);
    db.list();
}
```

老师，这个程序本来用Java写是在构造cd，dvd对象的时候调用super（title）去用基类的构造方法，查了下C++资料没找到super的用法，然后想到用组合构造，这样的实现和Java的super()用法是一样的吗，还是会有些出入什么的?

然后这个程序这样实际上会编译出错的，编译器说 Item没有默认的无参构造函数，手动加了个无参构造函数就能运行了，这是啥原理啊。按理说不是子类里面有父类完全一样名字的成员变量，父类的就会被隐藏么，为什么构造的时候还会先走一遍Item的初始化，然后再到DVD的初始化

稍微改动了下代码，试图使用DVD里的print函数，也出现了报错。

```
class DVD :public Item{
private:
    string title, director, comment;
    int  playingTime;
    bool gotIt = false;
public:
    DVD(string title, string director, int playingTime, string comment);
};
DVD::DVD(string title, string director, int playingTime, string comment){
    this->title = title;
    setTitle("b");
    this->playingTime = playingTime;
    this->comment = comment;
    this->director = director;
    }
    void print() {
        cout << "DVD:" <<  ":" <<  director << endl;
    }
```

未定义标识符"director"，就很奇怪啊，

然后改成

```
void print() {
        cout << "DVD:" <<  ":" <<  this.director << endl;
    }
```

也出现了报错： this 只能用在非静态成员函数内部?

可是我从头到尾都没定义过static欸

然后我又修改了一下print函数的代码

```
cout << "DVD:" <<  ":" <<  DVD::director << endl;
```

这个时候也报错，说成员 director被设为不可访问，就很奇怪啊，在DVD这个类里面，为什么他自己的私有成员不能在他自己的成员函数中被访问……

我就只想单纯打印一下director，到底该这么做呢……