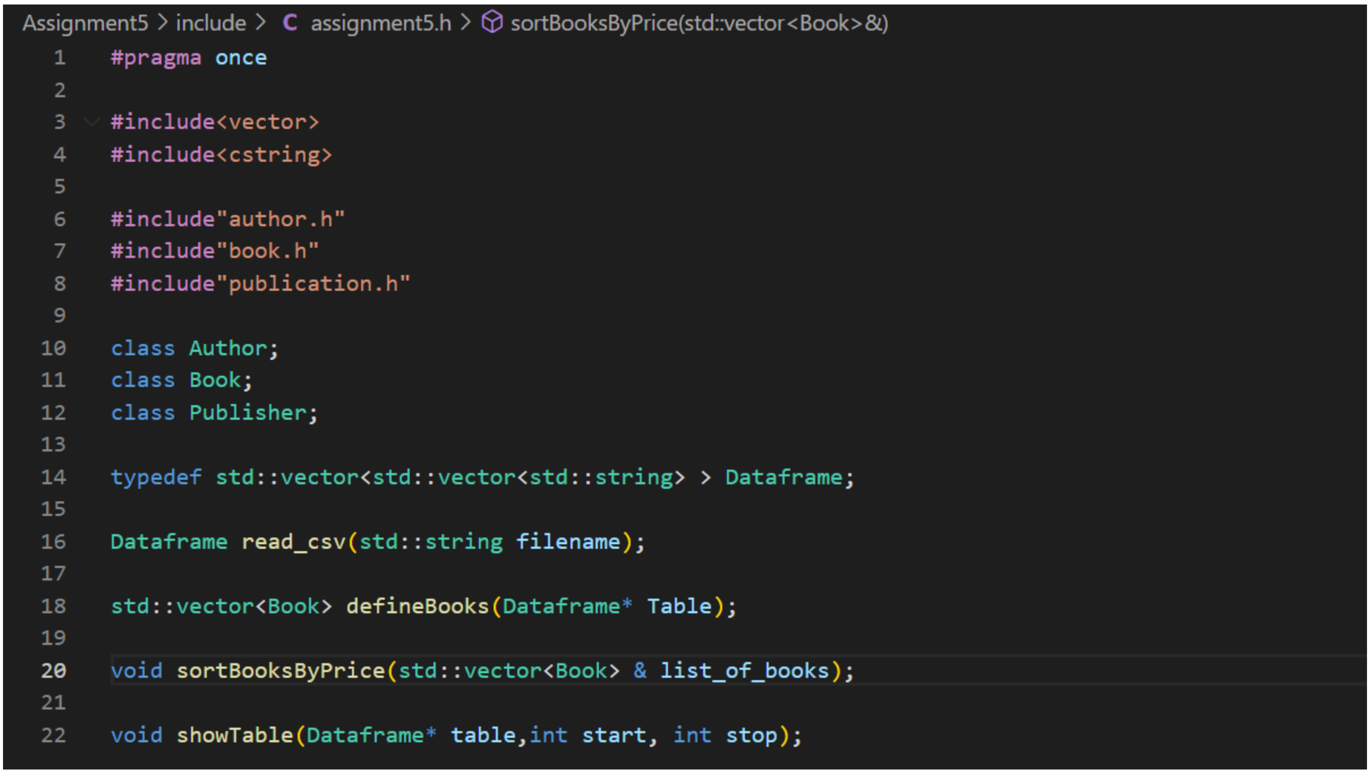
C++面向对象程序设计 作业报告5

报告人：邓雍　学号：2022141220184

目标实现：实现一个书籍的系统管理，其中包含作家信息分类管理，作家笔下所属书籍，书籍对应作家，对应出版社，出版社旗下作品。实现从csv中读取数据，将书籍按照价格升序排列。

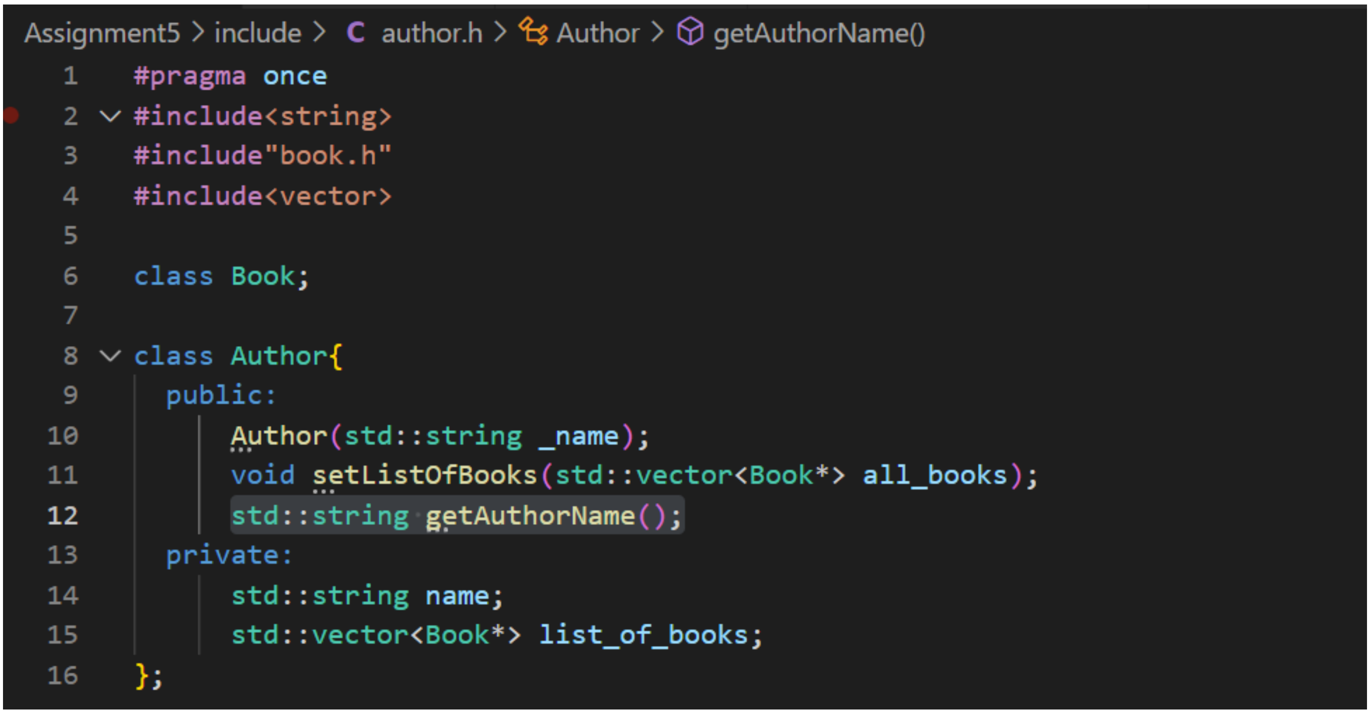
部分实现细节总结：由于存在嵌套编译问题，并不能直接按照题目中给出的模板实现。主要体现在不能使用&实现。更多地应用指针来解决这类问题。其次是readme中给出的结构体框架并不完整，仔细阅读测试代码后会发现需额外在book中申明两个public函数来访问内在变量。读入方面，由于书籍部分title，作者中带有','所以不能直接以其为分隔符，要做特殊的判断来添加。

下贴出实现代码。

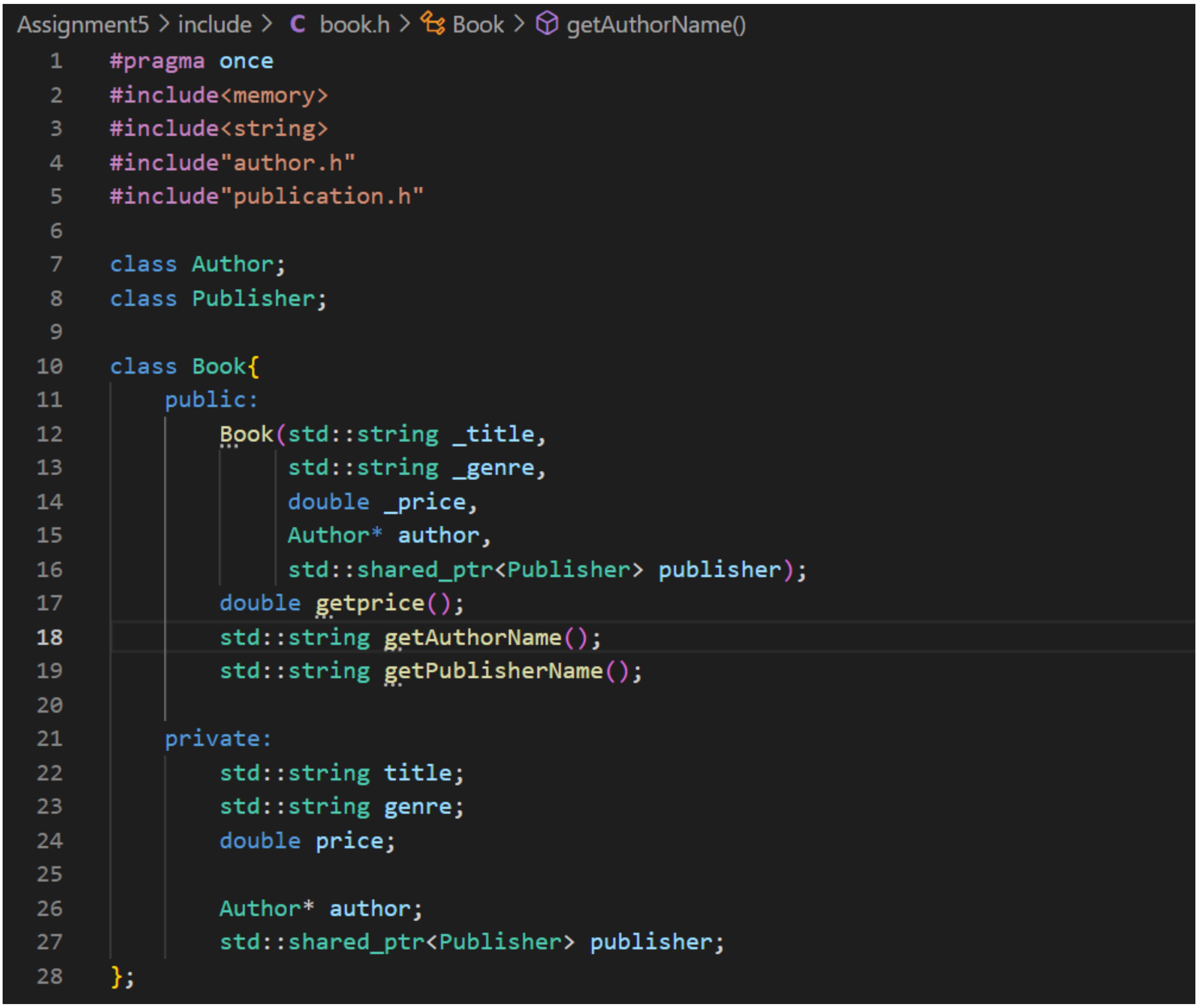

Assignment5 > include > C assignment5.h > sortBooksByPrice(std::vector<Book>&)

```cpp
#pragma once

#include<vector>
#include<cstring>

#include"author.h"
#include"book.h"
#include"publication.h"

class Author;
class Book;
class Publisher;

typedef std::vector<std::vector<std::string> > Dataframe;

Dataframe read_csv(std::string filename);

std::vector<Book> defineBooks(Dataframe* Table);

void sortBooksByPrice(std::vector<Book> & list_of_books);

void showTable(Dataframe* table,int start, int stop);
```

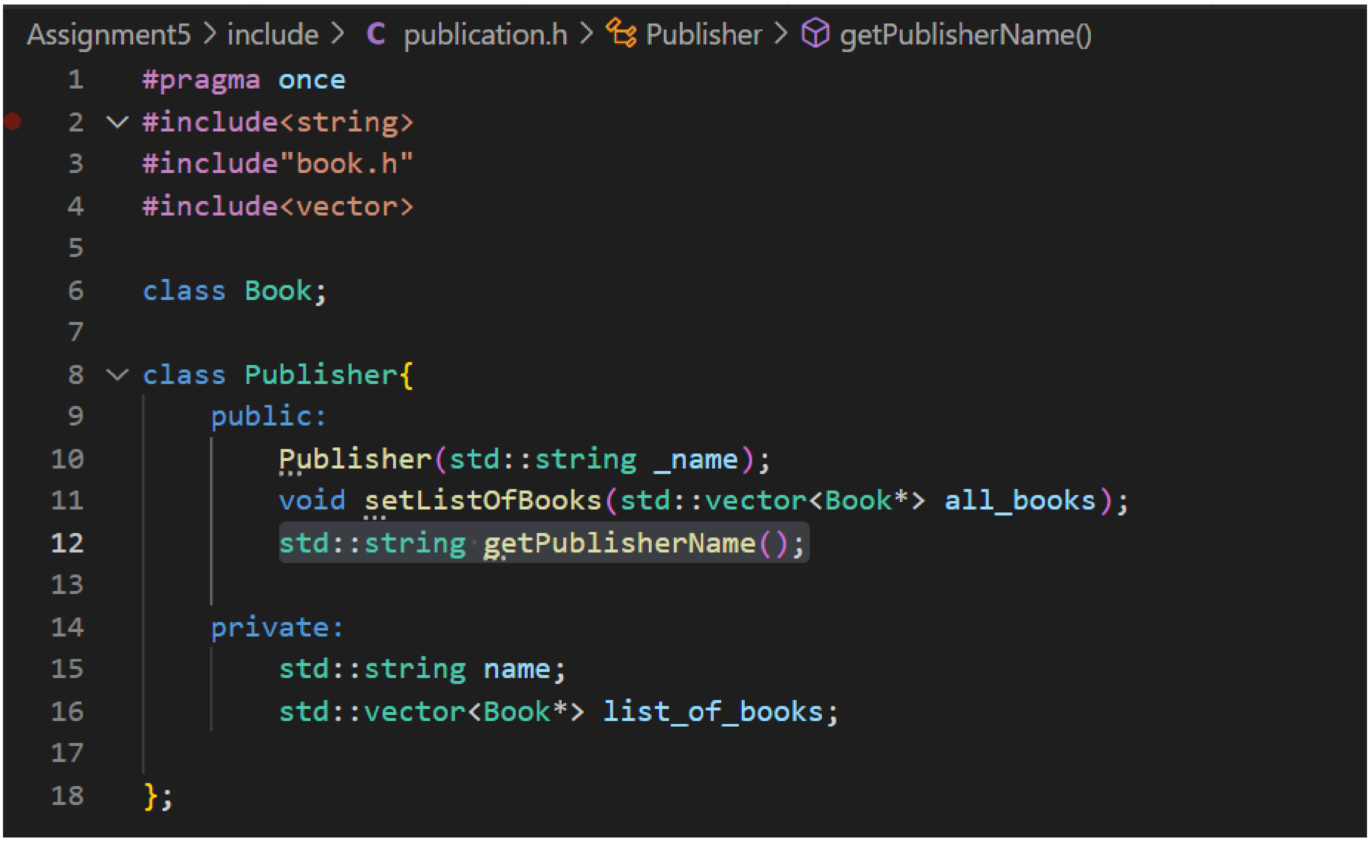
首先是assignment5.h的实现，无太多特别之处，可以在此进行统筹管理，声明三种类别及其他所需函数声明即可。


Assignment5 > include > C author.h > Author > getAuthorName()

```cpp
#pragma once
#include<string>
#include"book.h"
#include<vector>

class Book;

class Author{
  public:
      Author(std::string _name);
      void setListOfBooks(std::vector<Book*> all_books);
      std::string getAuthorName();
  private:
      std::string name;
      std::vector<Book*> list_of_books;
};
```



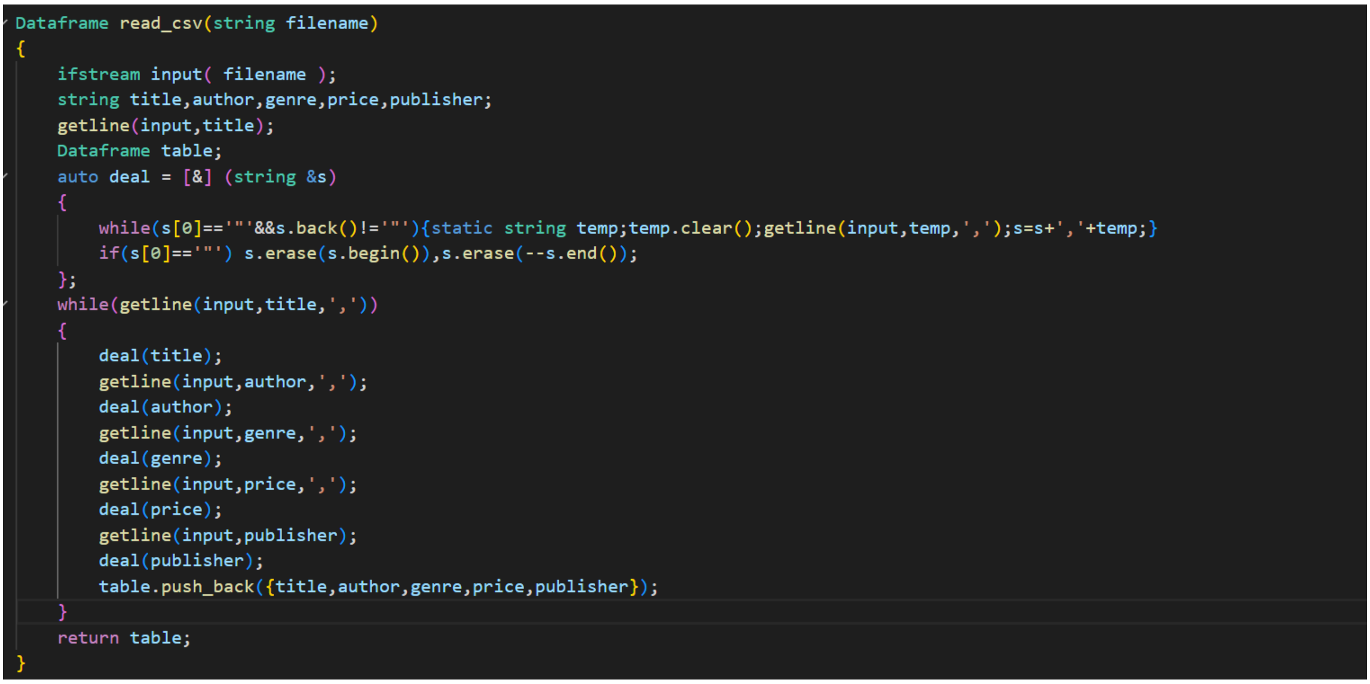
```cpp
#pragma once
#include<memory>
#include<string>
#include"author.h"
#include"publication.h"

class Author;
class Publisher;

class Book{
    public:
        Book(std::string _title,
             std::string _genre,
             double _price,
             Author* author,
             std::shared_ptr<Publisher> publisher);
        double getprice();
        std::string getAuthorName();
        std::string getPublisherName();

    private:
        std::string title;
        std::string genre;
        double price;

        Author* author;
        std::shared_ptr<Publisher> publisher;
};
```



Assignment5 > include > C publication.h > Publisher > getPublisherName()

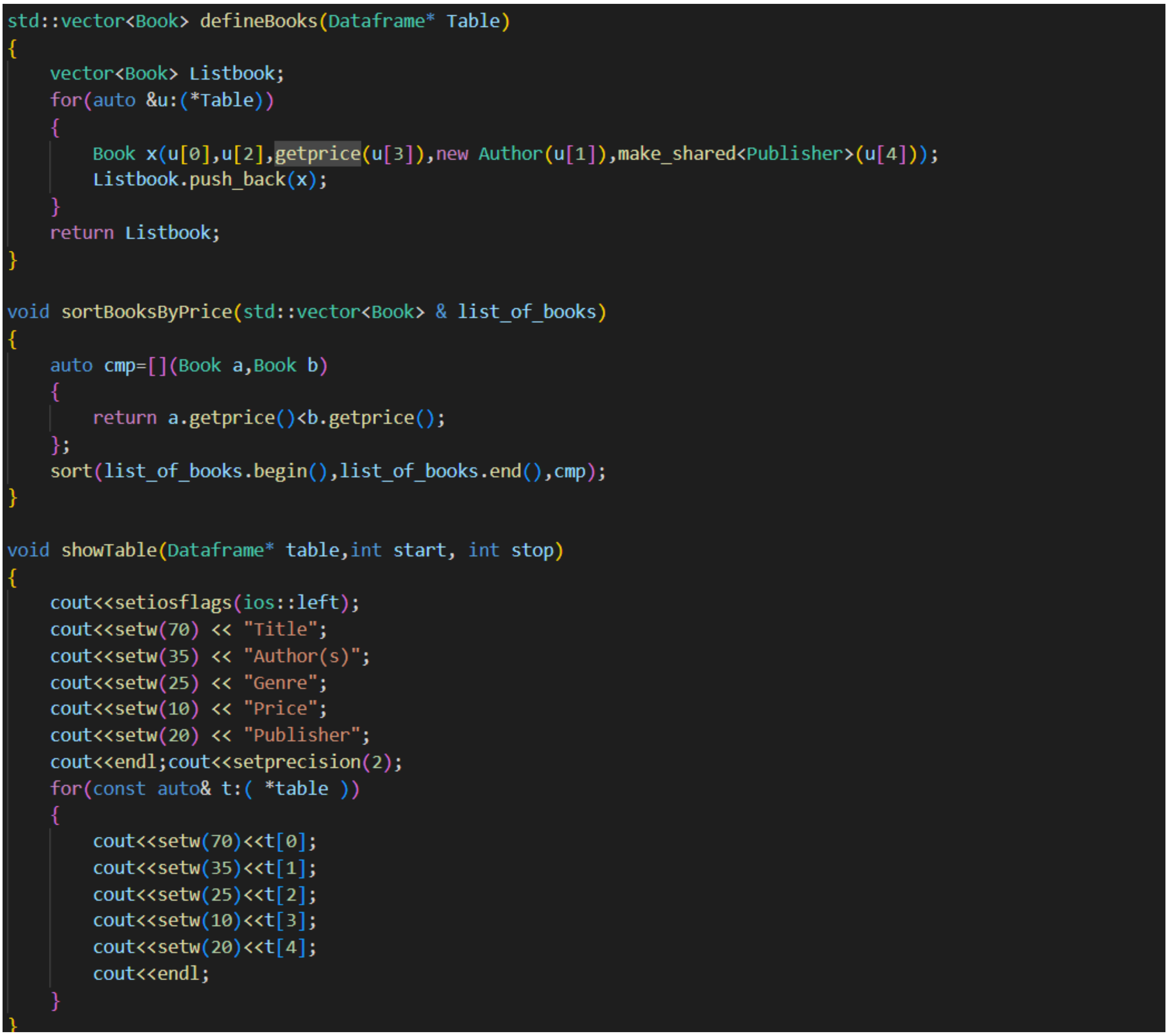
```cpp
#pragma once
#include<string>
#include"book.h"
#include<vector>

class Book;

class Publisher{
    public:
        Publisher(std::string _name);
        void setListOfBooks(std::vector<Book*> all_books);
        std::string getPublisherName();

    private:
        std::string name;
        std::vector<Book*> list_of_books;

};
```

本质无太多特别之处，按题目要求声明即可。注意额外声明所需的访问函数。以及将&改为*即可。注意声明可以重复，实现不要重复即可。



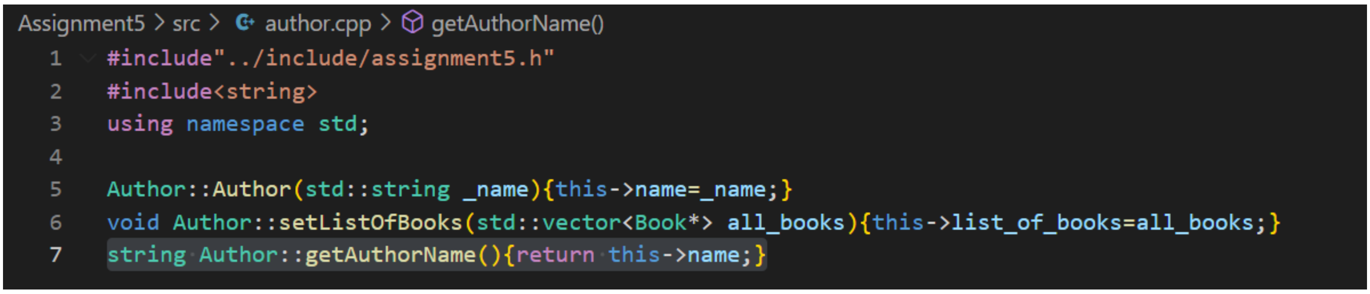
```cpp
Dataframe read_csv(string filename)
{
    ifstream input( filename );
    string title,author,genre,price,publisher;
    getline(input,title);
    Dataframe table;
    auto deal = [&] (string &s)
    {
        while(s[0]=='"'&&s.back()!='"'){static string temp;temp.clear();getline(input,temp,',');s=s+','+temp;}
        if(s[0]=='"') s.erase(s.begin()),s.erase(--s.end());
    };
    while(getline(input,title,','))
    {
        deal(title);
        getline(input,author,',');
        deal(author);
        getline(input,genre,',');
        deal(genre);
        getline(input,price,',');
        deal(price);
        getline(input,publisher);
        deal(publisher);
        table.push_back({title,author,genre,price,publisher});
    }
    return table;
}
```

read**函数的实现，实现一个算子来帮助我们解决带‘,’的标题问题，正则表达式匹配也是可选方法**

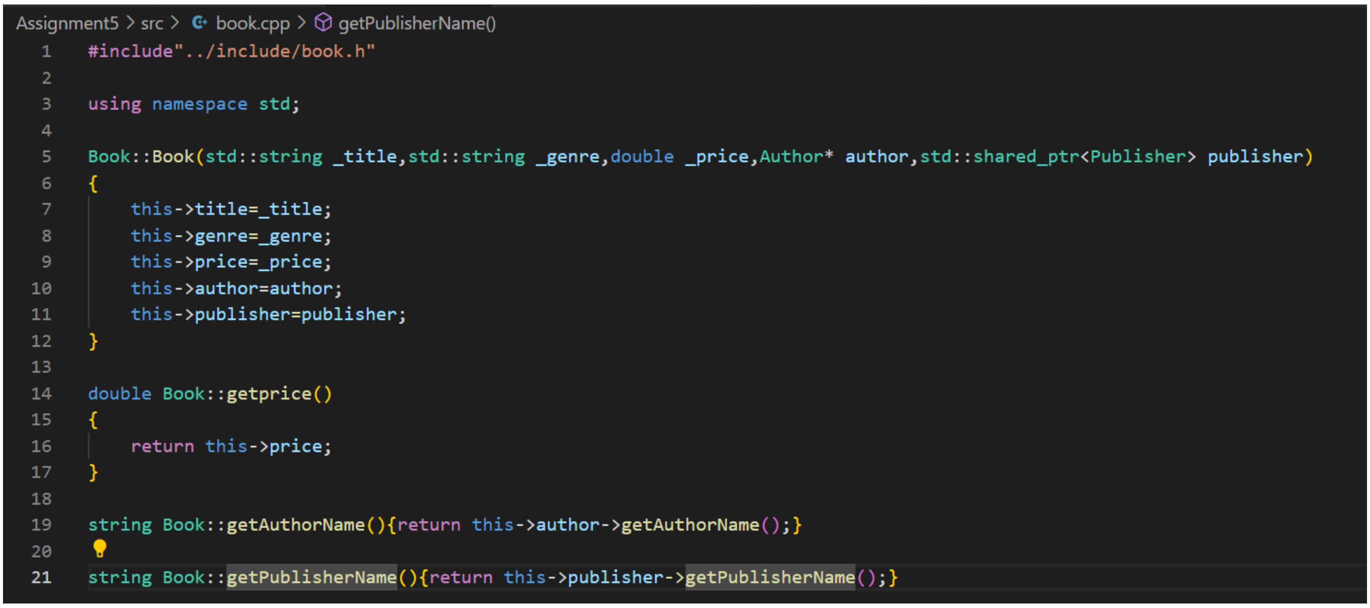


```cpp
std::vector<Book> defineBooks(Dataframe* Table)
{
    vector<Book> Listbook;
    for(auto &u:(*Table))
    {
        Book x(u[0],u[2],getprice(u[3]),new Author(u[1]),make_shared<Publisher>(u[4]));
        Listbook.push_back(x);
    }
    return Listbook;
}

void sortBooksByPrice(std::vector<Book> & list_of_books)
{
    auto cmp=[](Book a,Book b)
    {
        return a.getprice()<b.getprice();
    };
    sort(list_of_books.begin(),list_of_books.end(),cmp);
}

void showTable(Dataframe* table,int start, int stop)
{
    cout<<setiosflags(ios::left);
    cout<<setw(70) << "Title";
    cout<<setw(35) << "Author(s)";
    cout<<setw(25) << "Genre";
    cout<<setw(10) << "Price";
    cout<<setw(20) << "Publisher";
    cout<<endl;cout<<setprecision(2);
    for(const auto& t:( *table ))
    {
        cout<<setw(70)<<t[0];
        cout<<setw(35)<<t[1];
        cout<<setw(25)<<t[2];
        cout<<setw(10)<<t[3];
        cout<<setw(20)<<t[4];
        cout<<endl;
    }
}
```

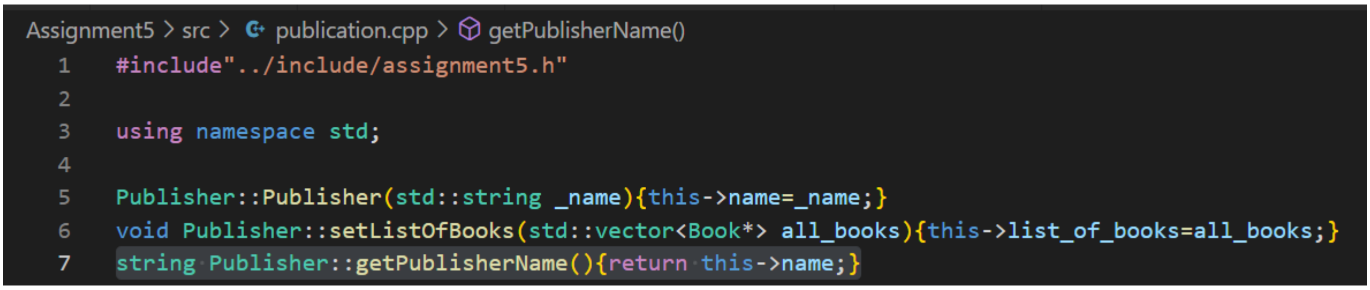
三个函数的实现，注意读入顺序与构造函数顺序不同即可。



```cpp
#include"../include/assignment5.h"
#include<string>
using namespace std;

Author::Author(std::string _name){this->name=_name;}
void Author::setListOfBooks(std::vector<Book*> all_books){this->list_of_books=all_books;}
string Author::getAuthorName(){return this->name;}
```

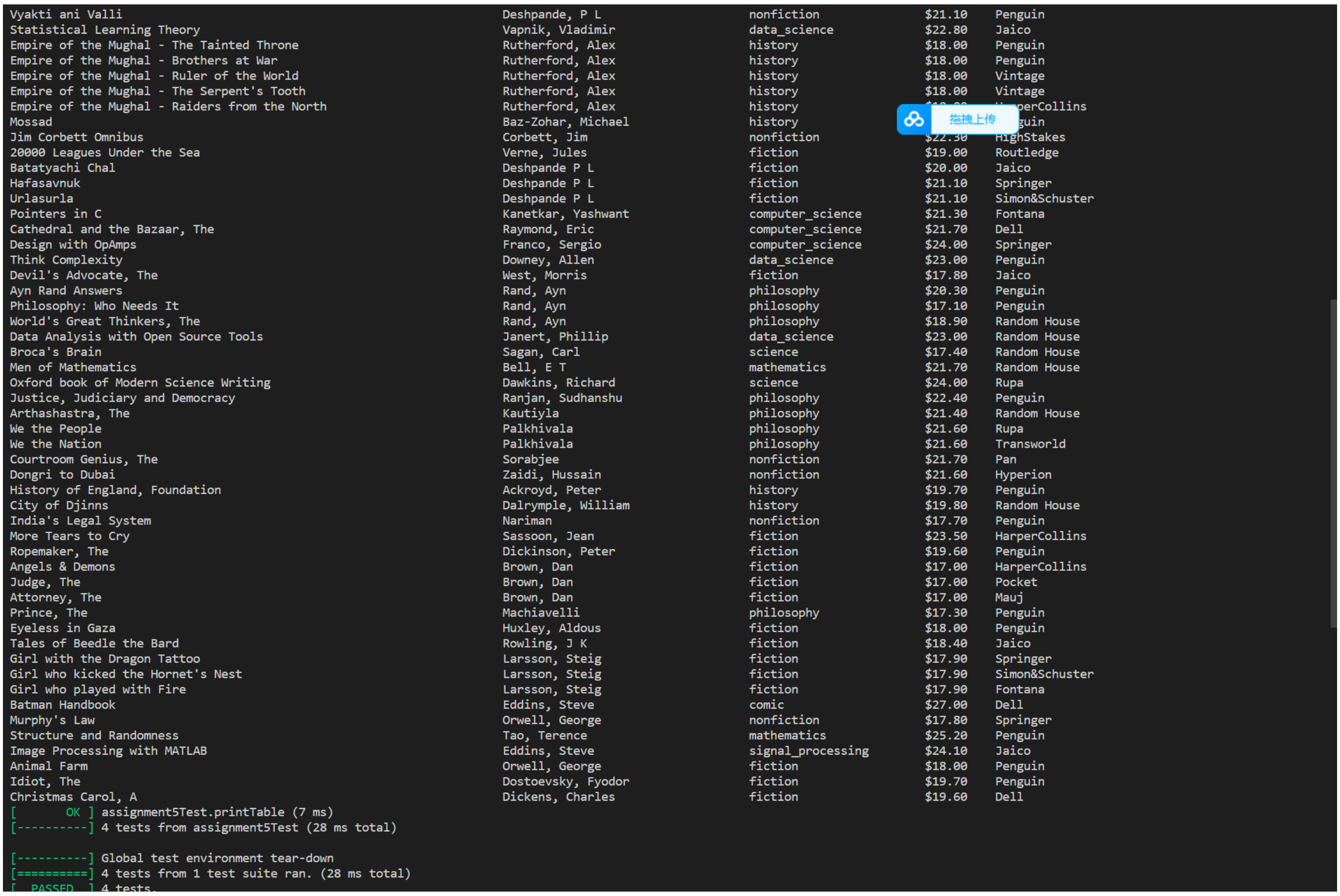


```cpp
#include"../include/book.h"

using namespace std;

Book::Book(std::string _title,std::string _genre,double _price,Author* author,std::shared_ptr<Publisher> publisher)
{
    this->title=_title;
    this->genre=_genre;
    this->price=_price;
    this->author=author;
    this->publisher=publisher;
}

double Book::getprice()
{
    return this->price;
}

string Book::getAuthorName(){return this->author->getAuthorName();}

string Book::getPublisherName(){return this->publisher->getPublisherName();}
```



```cpp
#include"../include/assignment5.h"

using namespace std;

Publisher::Publisher(std::string _name){this->name=_name;}
void Publisher::setListOfBooks(std::vector<Book*> all_books){this->list_of_books=all_books;}
string Publisher::getPublisherName(){return this->name;}
```

另外三个文件中主要是对函数中构造函数及部分赋值，调用函数的实现。



```
Vyakti ani Valli                                Deshpande, P L        nonfiction          $21.10   Penguin
Statistical Learning Theory                     Vapnik, Vladimir      data_science        $22.80   Jaico
Empire of the Mughal - The Tainted Throne       Rutherford, Alex      history             $18.00   Penguin
Empire of the Mughal - Brothers at War          Rutherford, Alex      history             $18.00   Penguin
Empire of the Mughal - Ruler of the World       Rutherford, Alex      history             $18.00   Vintage
Empire of the Mughal - The Serpent's Tooth      Rutherford, Alex      history             $18.00   Vintage
Empire of the Mughal - Raiders from the North   Rutherford, Alex      history             [illegible]   [illegible]perCollins
Mossad                                          Baz-Zohar, Michael    history             [illegible]   [illegible]guin
Jim Corbett Omnibus                             Corbett, Jim          nonfiction          $22.30   HighStakes
20000 Leagues Under the Sea                     Verne, Jules          fiction             $19.00   Routledge
Batatyachi Chal                                 Deshpande P L         fiction             $20.00   Jaico
Hafasavnuk                                      Deshpande P L         fiction             $21.10   Springer
Urlasurla                                       Deshpande P L         fiction             $21.10   Simon&Schuster
Pointers in C                                   Kanetkar, Yashwant    computer_science    $21.30   Fontana
Cathedral and the Bazaar, The                   Raymond, Eric         computer_science    $21.70   Dell
Design with OpAmps                              Franco, Sergio        computer_science    $24.00   Springer
Think Complexity                                Downey, Allen         data_science        $23.00   Penguin
Devil's Advocate, The                           West, Morris          fiction             $17.80   Jaico
Ayn Rand Answers                                Rand, Ayn             philosophy          $20.30   Penguin
Philosophy: Who Needs It                        Rand, Ayn             philosophy          $17.10   Penguin
World's Great Thinkers, The                     Rand, Ayn             philosophy          $18.90   Random House
Data Analysis with Open Source Tools            Janert, Phillip       data_science        $23.00   Random House
Broca's Brain                                   Sagan, Carl           science             $17.40   Random House
Men of Mathematics                              Bell, E T             mathematics         $21.70   Random House
Oxford book of Modern Science Writing           Dawkins, Richard      science             $24.00   Rupa
Justice, Judiciary and Democracy                Ranjan, Sudhanshu     philosophy          $22.40   Penguin
Arthashastra, The                               Kautiyla              philosophy          $21.40   Random House
We the People                                   Palkhivala            philosophy          $21.60   Rupa
We the Nation                                   Palkhivala            philosophy          $21.60   Transworld
Courtroom Genius, The                           Sorabjee              nonfiction          $21.70   Pan
Dongri to Dubai                                 Zaidi, Hussain        nonfiction          $21.60   Hyperion
History of England, Foundation                  Ackroyd, Peter        history             $19.70   Penguin
City of Djinns                                  Dalrymple, William    history             $19.80   Random House
India's Legal System                            Nariman               nonfiction          $17.70   Penguin
More Tears to Cry                               Sassoon, Jean         fiction             $23.50   HarperCollins
Ropemaker, The                                  Dickinson, Peter      fiction             $19.60   Penguin
Angels & Demons                                 Brown, Dan            fiction             $17.00   HarperCollins
Judge, The                                      Brown, Dan            fiction             $17.00   Pocket
Attorney, The                                   Brown, Dan            fiction             $17.00   Mauj
Prince, The                                     Machiavelli           philosophy          $17.30   Penguin
Eyeless in Gaza                                 Huxley, Aldous        fiction             $18.00   Penguin
Tales of Beedle the Bard                        Rowling, J K          fiction             $18.40   Jaico
Girl with the Dragon Tattoo                     Larsson, Steig        fiction             $17.90   Springer
Girl who kicked the Hornet's Nest               Larsson, Steig        fiction             $17.90   Simon&Schuster
Girl who played with Fire                       Larsson, Steig        fiction             $17.90   Fontana
Batman Handbook                                 Eddins, Steve         comic               $27.00   Dell
Murphy's Law                                    Orwell, George        nonfiction          $17.80   Springer
Structure and Randomness                        Tao, Terence          mathematics         $25.20   Penguin
Image Processing with MATLAB                    Eddins, Steve         signal_processing   $24.10   Jaico
Animal Farm                                     Orwell, George        fiction             $18.00   Penguin
Idiot, The                                      Dostoevsky, Fyodor    fiction             $19.70   Penguin
Christmas Carol, A                              Dickens, Charles      fiction             $19.60   Dell
[       OK ] assignment5Test.printTable (7 ms)
[----------] 4 tests from assignment5Test (28 ms total)

[----------] Global test environment tear-down
[==========] 4 tests from 1 test suite ran. (28 ms total)
[  PASSED  ] 4 tests.
```

附上测试点通过截图

makefile文件附在附件中