NAME: SATVIK DANDALE

DIV: TY E

BATCH: B1

GR NO: 1710797

ASSIGNMENT: STOCK CLASS

/\*

    Write a program to implement a class for book details having the data members book Id, Name, author, price, etc.

    The class has two public functions: getDetails and putDetails.

    Get details of some book from some users and print in tabular format. Also find the total price of all books.

    Write the program to extend the book class with dynamic memory allocation by using "new" keyword and object pointers.

    Write a Stock class which has a data member stock which stores the number of copies of each book, and total\_price which stores the total\_price of each type of book(may be more than 1)

\*/

#include<bits/stdc++.h>

using namespace std;

#define id\_width 6

#define name\_width 20

#define author\_width 20

#define price\_width 10

#define stock\_width 5

#define total\_price\_width 5   //Just use Total as the column name

class Book{

    protected:

        int id;

        float price;

        string name, author;

    public:

        Book(){

            id = 0;

            price = 0;

            name = "";

            author = "";

        }

        Book(int id, float price, string name, string author){

            // Paremeterized contructor

            this->id = id;

            this->price = price;

            this->name = name;

            this->author = author;

        }

        void setDetails(){

            cout<<"\nEnter the Book Id: ";

            cin>>this->id;

            cout<<"Enter the Price of this book: ";

            cin>>this->price;

            cin.ignore();   // Ignore the garbage remaining in the istream.

            cout<<"Enter the Name of the book: ";

            getline(cin, this->name);

            cout<<"Enter the name of the Author of this book: ";

            getline(cin, this->author);

        }

        int getId(){return this->id;}

        float getPrice(){return this->price;}

        string getName(){return this->name;}

        string getAuthor(){return this->author;}

        void putDetails(){

            cout<<setw(id\_width)<<this->id;

            cout<<setw(name\_width)<<this->name;

            cout<<setw(author\_width)<<this->author;

            cout<<setw(price\_width)<<this->price<<endl;

        }

};

class Stock: public Book{   //Inheriting from Book

    private:

        int stock;

        float total\_price;

    public:

        Stock(){

            this->stock = 0;

            this->total\_price = 0;

        }

        Stock(int id, float price, string name, string author, int stock):Book(id, price, name, author){

            this->stock = stock;

            this->total\_price = this->price \* this->stock;

        }

        void setDetails(){  // OVERRIDING THIS FUNCTION FROM THE BASE CLASS

            cout<<"\nEnter the Book Id: ";

            cin>>this->id;

            cout<<"Enter the Price of this book: ";

            cin>>this->price;

            cout<<"Enter the stock of this book: ";

            cin>>this->stock;

            this->total\_price = this->price \* this->stock;

            cin.ignore();   // Ignore the garbage remaining in the istream.

            cout<<"Enter the Name of the book: ";

            getline(cin, this->name);

            cout<<"Enter the name of the Author of this book: ";

            getline(cin, this->author);

        }

        void putDetails(){  // OVERRIDING THIS FUNCTION FROM THE BASE CLASS

            cout<<setw(id\_width)<<this->id;

            cout<<setw(name\_width)<<this->name;

            cout<<setw(author\_width)<<this->author;

            cout<<setw(price\_width)<<this->price;

            cout<<setw(stock\_width)<<this->stock;

            cout<<setw(total\_price\_width)<<this->total\_price<<endl;

        }

        friend float getGrandTotal(Stock \*, int);

};

float getGrandTotal(Stock \*list, int count){

    int grandTotal = 0;

    for(int i=0; i<count; i++)

        grandTotal += list[i].total\_price;

    return grandTotal;

}

int main(){

    Stock \*p;

    int n;

    cout<<"Enter the number of books\n";

    cin>>n;

    p = new Stock[n];

    for(int i=0; i<n; i++){

        cout<<"\nEnter the details for book no "<<i+1<<":\n";

        p[i].setDetails();

    }

    cout<<setw(id\_width)<<"BookId";

    cout<<setw(name\_width)<<"Book Name";

    cout<<setw(author\_width)<<"Book Author";

    cout<<setw(price\_width)<<"Book Price";

    cout<<setw(stock\_width)<<"Stock";

    cout<<setw(total\_price\_width)<<"Total Price"<<endl;

    for(int i=0; i<n; i++)

        p[i].putDetails();

    cout<<"The Grand Total Amount is: ";

    cout<<getGrandTotal(p, n);

}

OUTPUT:

Enter the number of books

2

Enter the details for book no 1:

Enter the Book Id: 1

Enter the Price of this book: 15

Enter the stock of this book: 3

Enter the Name of the book: B1

Enter the name of the Author of this book: A1

Enter the details for book no 2:

Enter the Book Id: 2

Enter the Price of this book: 25

Enter the stock of this book: 2

Enter the Name of the book: B2

Enter the name of the Author of this book: A2

BookId Book Name Book AuthorBook PriceStockTotal Price

1 B1 A1 15 3 45

2 B2 A2 25 2 50

The Grand Total Amount is: 95