Serendipity Booksellers Software Development Project— Part 15: A Problem-Solving Exercise

1. Break the `BookData` Class into Two Classes

Currently the `BookData` class contains all the data about a book in the store's inventory. Now you will break the class into two classes: one that is a base class containing general data about a book, and another that is a derived class containing data about a book in inventory.

First, simplify the `BookData` class so it contains only the general data about a book. Specifically, modify the `BookData` class so it contains only the following member variables and member functions:

Member Variables

```
bookTitle
isbn
Author
Publisher
```

Member Functions

```
setTitle
setISBN
setAuthor
setPub
```

Next you will create a new class named `InventoryBook`. This class will be derived from the `BookData` class, and will hold inventory-related data about a book. Specifically, this class will contain the following member variables and member functions, which were removed from the original `BookData` class:

Member Variables

```
dateAdded
qtyOnHand
wholesale
retail
```

Member Functions

```
setDateAdded

setQty
```

```
setWholesale
setRetail
isEmpty
removeBook
```

2. Create the SoldBook Class

Create a class named `SoldBook`, which is derived from the `InventoryBook` class. Its purpose is to represent a book that has been sold to a customer, and perform the necessary calculations for the sale of a book. It should have the following members:

Member Variables:

`taxRate`	A private static member, used to hold the sales tax rate.
`qtySold`	The quantity of a particular book that is being purchased.
`tax`	The sales tax on the purchase of a particular book, calculated as `qtySold` times `retail` times `taxRate`. (`retail` is inherited from `InventoryBook`.)
`subtotal`	The subtotal of the sale of a particular title. The subtotal is calculated as `retail` times `qtySold` plus `tax`.
`total`	A private static member used to hold the total of an entire sale.

You should determine the accessors, mutators, and other member functions needed in this class.

The cashier function should ask the user how many tittles the customer is purchasing. It should then dynamically allocate an array of `SoldBook` objects large enough for that many titles. The function will use the array of `SoldBook` objects to compute the necessary information for a customer's sale. The function will then display the simulated sales slip on the screen.