COP 4814 Lab: BookFinderService, Part 2

This lab continues work you began during the previous Web services lab.

Step 1. Define a class named Book in your IService.cs file by changing the class name to *Book*. Add properties to the class for the ID (string), Title (string), Author (string), and Year (int). Use the DataContract and DataMember attributes to make the class and its properties visible to client programs.

Step 2. Modify your constructor in Service.cs to build a list of 10 books (having the properties listed in Step #1).

Step 3. Modify the GetBookList service method so it returns a sorted array of Book objects. The calling program indicates which field should be used for the sort, by passing a string parameter equal to one of the following single letters: ("I" = ID, "T" = title, "A" = author, "Y" = year).

In the Web service, define a separate class for each sort field. Each class implements the IComparer interface, as shown in this sample for the ID sort field:

```
public class BookIdComparer : IComparer<Book>
{
  int IComparer<Book>.Compare(Book x, Book y)
  {
     return x.ID.CompareTo(y.ID);
  }
}
```

And this is how you would sort the list by ID:

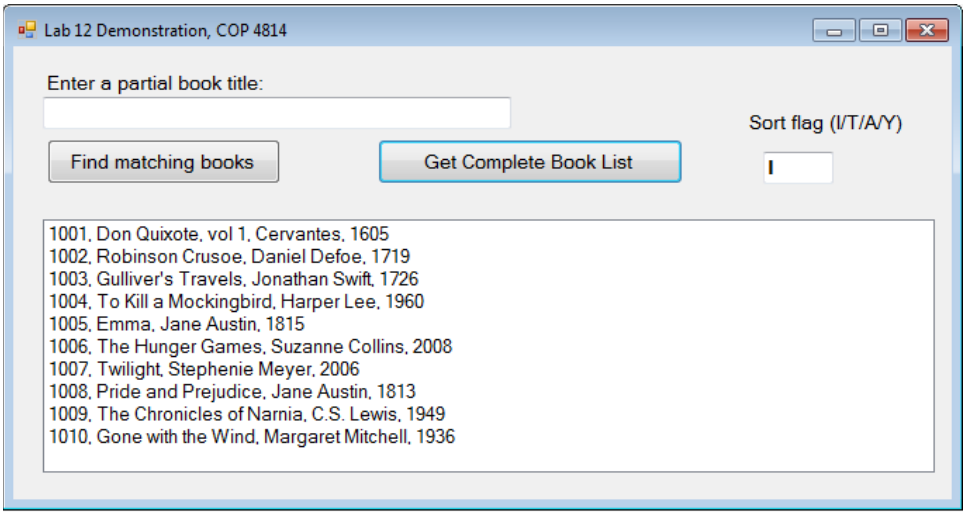
```
bookList.Sort(new BookIdComparer());
```

You can read much more about the IComparer interface on the Microsoft web site.

The Book class, because it is defined as a DataContract inside the web service, could contain a ToString method, but there's no way to make that method visible to the client program. On the other hand, you can add another method to the BookService web service named **ToBookString(Book b)** that returns a string representation of a book. Then you can call it from the client program, passing it a Book object.

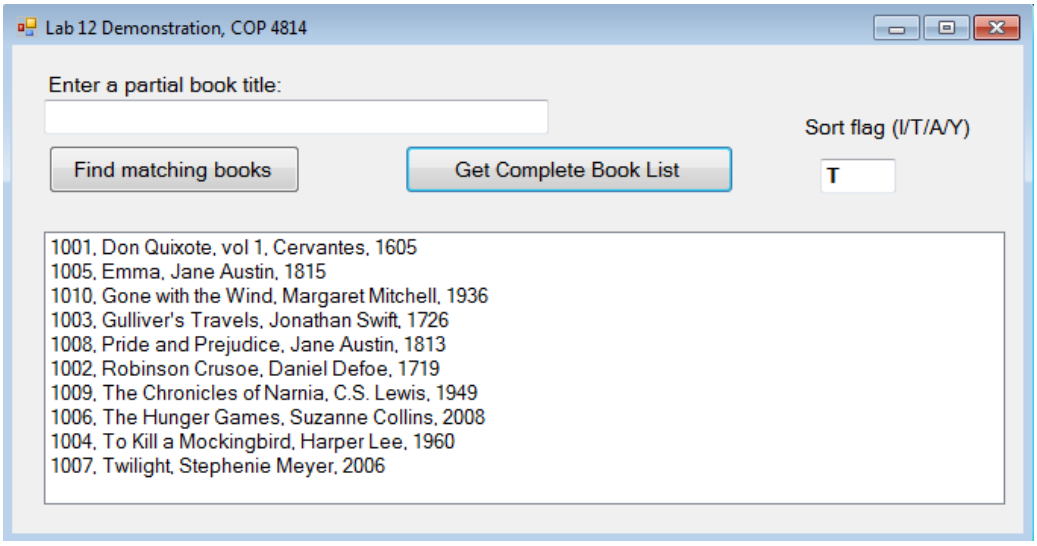
Step 3: Update the Client Program

1. Change the user interface to this:

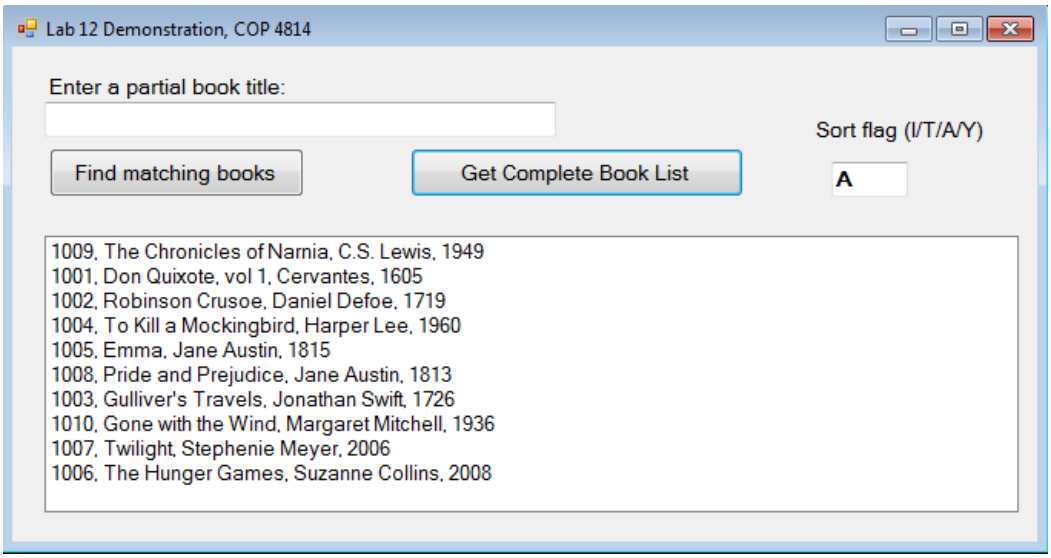


2. The G*et Complete Book List* button calls a web service method to get a list of Book objects. Display the list of books in a ListBox control.

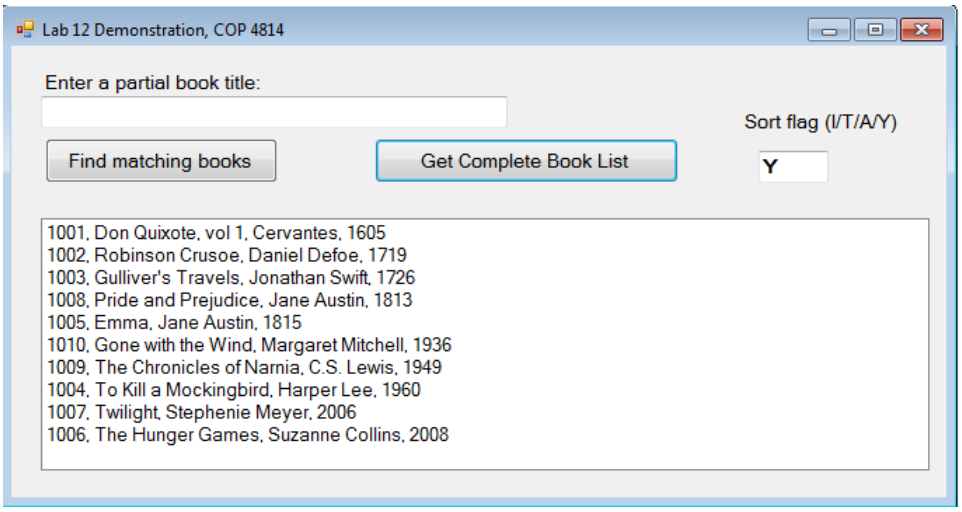
3. Let the user enter a capital letter into the textbox, to specify the sort order of the books ("I" = ID, "T" = title, "A" = author, "Y" = year). Pass this letter to the GetBookList method. Here, for example, the books are sorted by title:



And now they are sorted by author name:



And finally, by year:



4. The *Find Matching Books* button calls the web service method that tries to match a partial string with each of the book titles. The comparisons must be case-insensitive. Here's an example: