Start with this project that we went over in class:   [4-ToDoLL-WorkItemOrdIndSol.zipView in a new window](https://bc.instructure.com/courses/1262746/files/65536300/download?wrap=1) Modify the Link List code, and the Form code to convert the program to be a program to store books you want to read, stored by a book review rating that you enter.

=======================================================================================

1- Start by creating a Book class. This class should have these properties:

Title

Author

Rating  (1 is a great book, down to a 5, which is a book not worth reading)

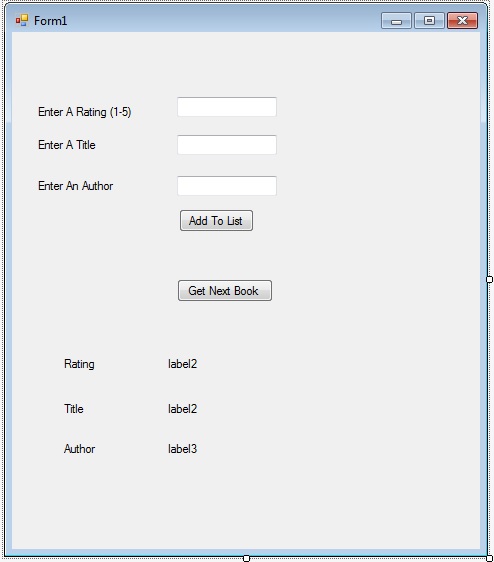
Mark the class public!!!

2- Then modify your Form code to accept these 3 values, so a user can submit a new book, and retrieve one with the highest rating in the list and see these 3 values displayed at the bottom of the form. Note that since we are only allowing 5 rating levels, if a user submits, for example, books with these title-ratings in this order:

A-3 B-1 C-5 D-3 E-4 F-1 G-3 H-5 and then retrieves them, they will get them back in this order:

F-1 B-1 G-3 D-3 A-3 E-4 H-5 C-5 So they will come out ***by priority grouping***, but within a given priority, they will come out in the opposite order they were inserted.

Edit the form so that you have 3 text boxes, and you should change the **label’s text displayed** as well as the C# **property name for the textboxes and labels** so they make sense. Here is what my form looks like after I made those changes:



(3) Now modify the C# code in the form to account for those name changes and deletions. (If you changed these using the property window on the form page, Visual Studio will already have changed most of the name in the C# code for you.)

Remove the references to the WorkItem and instead, instantiate a Book object and load its properties from the textboxes as makes sense.

We will be using a method called

ToDoLL.InsertInOrder(aBook);

Since we want to store by value (rating), and not by index.

Also modify the code that retrieves the next book to make the correct method call and fill in the correct label values.

(4) Now go modify the WorkItenLinkedList class to be able to properly handle books. Since we will not be using the

public void InsertAt(WorkItem pWorkItem, int index)

method, you should delete it from this project.

You need to find all references to the WorkItem class, or objects instantiated from it, and their properties, and make “corrections” so that this Linked List will store and retrieve Book objects instead of WorkItem objects.

The name of the file and the class (WorkItenLinkedList ) is probably not worth changing, as there is a high risk of Visual Studios source control and build code getting all confused.

Here is a screen shot after retrieving a book and about to enter another one:

