

# Object Oriented Programming

Workshop - Book with Page Turning

William L. Honig, Ph.D.

Create a full OOP Class

Use and test your Class

## Step 0

**Your Topic:** Remember the Book example (Book objects have author, title, publication year, total pages, and current reading position). Now, let's extend that by adding page turning.

**What determines the State** (internal data) for objects of this type? In addition, how could you store a bunch of pages, each page a String of text?

**What operations** / manipulations / access / transformations would a program need to do on these kinds of things? Think beyond simple set and get of the state data. Before we had the ability to change the current reading position (and make sure it makes sense). Need to be able to retrieve the current page of the book, in a form suitable for printing or other operations. It would also be nice to be able to create a book with an array of Strings, one per page of the book (in this case the total pages would match the size of the array).

Done? Create a UML Diagram or at least generate the words in a the comments of your java code.

## Step 1

**Create a class:** write some oop code. It should have (in order)

1. Some constructors (including a default constructor)
2. Suitable data attributes, each with a useful type
3. Several methods or operations. Start with just the signatures. Include toString, equals, and possibly compareTo.

Done? You are done (for now) if you have something that compiles with proper OOP structure. No need to have any body in the methods. All data attributes should be private. Make certain the parameters for each method make sense (don't miss any information you need to complete the method's job). Methods should be public if you want users of the class to use them, but private if they are internal helper methods.

## Step 2

Use and test your class: create a separate text class, containing main. Use it to

1. Create 3 or more objects of your class
2. Prompt the user (or use code) to get values and/or instructions for what to do with your objects.
3. Print out information about your objects.

There should be NO prompts for information from the user anywhere but in your test code. No getting input from user inside the class.

Done? You are done if main tests all parts of your class (even if the results don't always make sense). Create Book objects using all versions of your constructors. Call each method. The results may not be very pretty and may not make sense but it should run.

## Step 4

Refine your class: improve the class structure using good OOP. You may need to consider any of these

1. Improve the methods so they are more realistic
2. Add additional information to the object state information (and modify the methods as necessary to support them)
3. Create helper methods inside the class for your own use.

In general, methods should be simple, concise and “obviously correct”. Ask someone to read it. If too long, split it up. For constructors, use other constructors and provide suitable information from parameters or default values.

Done? Can you retrieve any page of the book and print it out from main? Can you create books in all possible ways? Does your code in the class keep the state correct at all times (be sure the current page makes sense)?

v1.0 (Original) 10 April 2018

©Dr. William L. Honig, Loyola University Chicago