COMP 1409 – Assignment #2 (50 points)
Due: 11:59 p.m. the night before session 5

For this assignment you will modify the Book class you wrote for the first assignment. Add two fields: quantity on order and quantity sold. Do not add any other fields.

- Provide accessor methods for both new fields.
- Provide a mutator method `setQuantityOnOrder()`that passes in the quantity to order as a parameter and adds that value to the current quantity on order. Be sure the quantity on order can never go below zero. Display an error message if the parameter is negative. Comment the decision.
- Modify the constructor to set both quantity on order and quantity sold to zero when a Book object is created.
- Modify the constructor to be sure that year published and purchase price are not negative values. You don't know how to prevent an object from being created if given invalid data, so use these as defaults in that situation: year: 2006, purchase price: 0.0.
- Modify the `restock()`method so that it will not allow the quantity in stock to be set to a negative value, and will display an error message if the parameter is negative. Comment the decision.
- Modify the `restock()` method. If the number of books to be added to inventory is greater than the number on order, the quantity in stock will be increased by only the amount on order and a message will be displayed that reports the discrepancy. Quantity on order will be reduced by the appropriate value.
- Modify the `sell()`method so that it will not allow the quantity in stock to be set to a negative value, and will display an error message if the parameter is greater than the quantity in stock or if the parameter is negative. Comment the decisions.
- Modify the `sell()` method so that it will not allow a book to be sold if the purchase price is set to zero or if the selling prices is set to zero. The method will display an error message if it is unable to sell the book. Comment the decision.
- Modify the `sell()`method to update the quantity sold by the appropriate amount.
- Modify both `setPurchasePrice()` and `setSellingPrice()` to ensure neither price can be below zero. The methods will display error messages if unable to set the relevant price. Comment the decisions.
- Add a method called `calcProfit()` that calculates and returns the profit from selling one copy of the book, based on the purchase price and selling

price. This method will not display anything on the screen. Comment the calculation.

- Add a method called `calcTotalProfit()` that calculates and returns the profit from all the books that have been sold. This method must invoke `calcProfit()`.
- Add a method called `calcInventoryValue()` that calculates and returns the current value of the quantity in stock, based on purchase price.
- Modify the `printDetails()` method to include information about the number of books on order, the number sold, the inventory value and the profit. This method must invoke the relevant “get” and “calc” methods.

  ```
  Title: All the Ghosts
  Author: Jack Ghostwriter
  Publisher: Spectre Press
  Year: 2005
  Purchase price: $12.43
  Selling price: $17.95
  Quantity in stock: 34
  Quantity on order: 65
  Quantity sold: 12
  Inventory value: $422.62
  Profit on sales: $66.24
  ```
- Be sure to update your constructor and method comments to reflect the changes you have made.
- Test your methods thoroughly to be sure everything works as specified.

Marks will be given for:

- Comments – appropriate and complete.
- Style – see the style guide Appendix J in your textbook.
- Correctness and completeness – code meets the requirements listed above.

Create a .zip file containing your entire BlueJ project (zip the folder, not the individual files). Name the .zip file with your name and the assignment number, e.g. “Susan_Wong_Assign_2.zip”. Upload the file to WebCT before the cutoff time.