**C# Project 3**

Create a Windows Form application for “The Cookie Source” that accomplishes the following:

* Instantiable class called Order to hold the order information when the user places an order:
  + Add the necessary details (properties, constructor(s), methods) to the Order class
  + An order should contain:
    - order number (auto-generated based on highest number in file)
    - customer name (required)
    - phone number (required)
    - cookie type (C – Chocolate Chip, O – Oatmeal, S – Sugar)
    - quantity (number of dozens; must be > 0)
    - order date (use current date)
    - delivery date (use DatePicker control; make sure user can’t select a date prior to order date)
  + Implement the IComparable interface so that orders can be sorted by delivery date.
  + Demonstrate the use of custom Exceptions by creating an Exception class for use in validation of one of the following: customer name, phone number, or quantity. Be sure to use this custom exception when validating the user input.
* User interface requirements:
  + DO NOT SORT MASTER LIST! MOVE TO TEMP LIST FOR DISPLAY/SORT
  + DISPLAY ORDERS IN SHOW DIALOG
  + Design of the GUI is your choice. Adhere to good design principles. Consider using multiple types of controls: text box, radio button, date pickers, etc.
  + At application start up (form load), read all order objects from the input file and place into a list of orders. This requires de-serialization. Remember, the very first time the project is ran, there will not be a file yet. Be sure to check for this to avoid any exceptions.
  + When the user adds a new order, add this order object to the list.
  + At application end (user exit), write all objects from the list back out to the file. This requires serialization. Do not append to the file, but write over what’s there.
  + Allow the user an option to display all the orders in the list on a different screen. Your choice as to how to display this information to the user. However, Orders should display in ascending delivery date order (oldest date first).
  + User should be able navigate between adding and displaying orders.
* Unit Test requirements:
  + Unit Test project should be located inside the project solution
  + create unit test methods to test the properties/constructors

Project is worth 100 points.