COSC 1P03 Assignment 5

***Due: Apr. 10, 2013 @ 12:00 pm noon** (late date Apr. 13)*

In preparation for this assignment, create a folder called `Assign5` with a subfolder: `Spartan` to contain the projects for the assignment. The files for this assignment can be downloaded from URL: `http://www.cosc.brocku.ca/Offerings/1P03/A5Files.zip`.

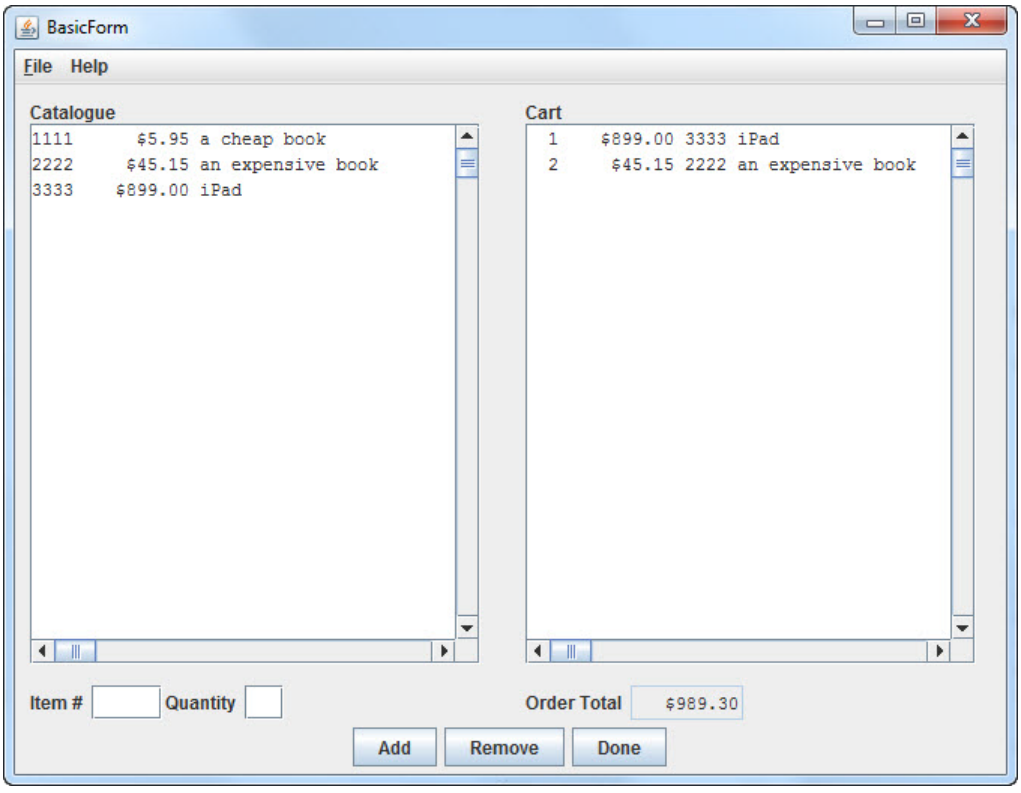
Problem

Spartan.com is a new on-line retail store planning to go "head-to-head" with Amazon.com. The store offers a number of items for sale. Each item has an item number, description and price. Customers of the store can order these items on-line. Once a customer's identity is validated by entry of userid and password, a cart (a collection of items that are being ordered) is established for the customer and the customer can begin ordering items. The customer selects the item by item number and specifies a quantity (number of items) that they wish to order. The customer can order additional quantity of an item, remove some number of a particular item from the order and order other items. Once the customer has completed the order, the order (cart) is confirmed, with a confirmation sent to the customer. At any time, a customer may have a number of orders that have been confirmed but not shipped.

Throughout the workday, confirmed orders are processed, on a first-come-first-served basis, for shipping. The shipping clerk examines the next order to be shipped, packs the items and then ships them to the customer's address. A shipping notice is also sent to the customer and the order is then considered complete.

Inputs & Outputs

A customer should be presented with a form like the one below when placing an order (i.e. after confirmation of his/her userid/password.)



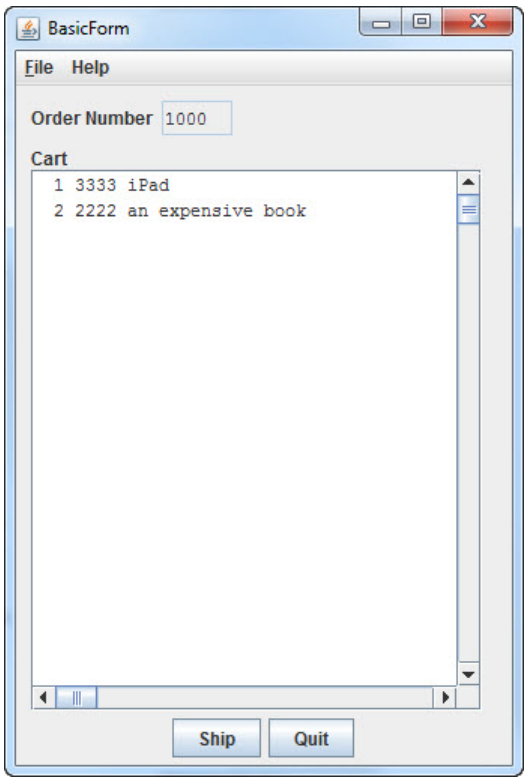
The `Catalogue` is a complete list of all the items the store has for sale. The `Cart` is the list of items that the customer has ordered, the first number on each line being the quantity. The `Item #` field is used by the customer to select an item from the catalogue. The `Quantity` field is used to specify the quantity. `Add` adds (an additional) quantity of the specified item. `Remove` removes (reduces) the quantity of the specified item from the order. Note that there should only be one entry for each item in the cart. Adding or removing some quantity of that item should change the quantity value. The customer presses `Done` when s/he has finished selecting items. Note that an empty cart does not result in an order actually being placed.

When the order is confirmed (i.e. after the customer has pressed `Done` on a non-empty cart), an order confirmation report such as below should be printed to be mailed to the customer.

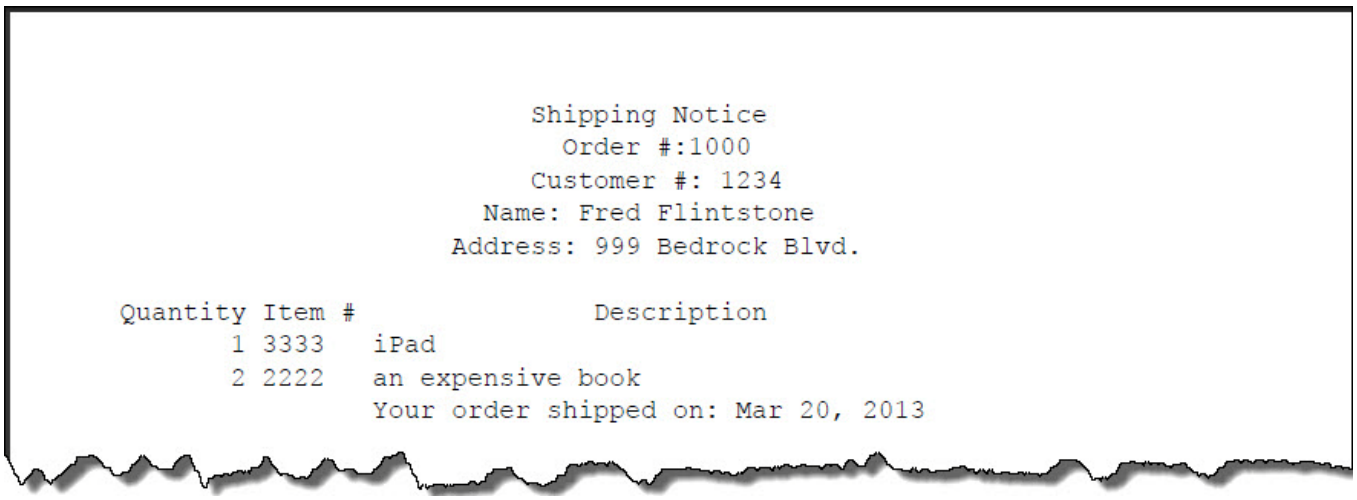
```
                        Order Confirmation
                           Order #:1000
                         Customer #: 1234
                       Name: Fred Flintstone
                     Address: 999 Bedrock Blvd.

Quantity Item #              Description                 Amount    Total
       1 3333   iPad                                    $899.00    $899.00
       2 2222   an expensive book                        $45.15     $90.30
                Total                                                $989.30
```

The shipping clerk should be presented with a form like the one below when processing orders for shipping.



The `Order Number` is the order number assigned to the order (cart) when the customer confirmed (completed) the order (note it matches the number on the confirmation report). The `Cart` is the contents of the cart (order) as completed by the customer, the first number on each line being the quantity. The clerk presses `Quit` to finish filling orders for the day. (The order displayed is **not** shipped.) The clerk presses `Ship` after packing up the requested items. A shipping notice (as seen below) is printed and mailed to the customer. The order is then complete and the next order pending shipping (if any) is presented on the form.



```
                        Shipping Notice
                          Order #:1000
                        Customer #: 1234
                     Name: Fred Flintstone
                   Address: 999 Bedrock Blvd.

Quantity Item #              Description
       1 3333   iPad
       2 2222   an expensive book
                Your order shipped on: Mar 20, 2013
```

Future Considerations

In future, it will be desirable to allow a customer to check their pending orders (i.e. orders that have been confirmed but not shipped.) Currently billing is done manually, the printing of an order confirmation informing the Accounts Receivable Department to initiate billing. In future it would be desirable for a customer's account to be tracked by the system (i.e. amount owed, billing notices, and receipt of payments). Also, it will be necessary to be able to add/remove items from the catalogue, add/remove customers from the customer list, and allow customers to change their password.

Requirements

1. Perform analysis as discussed in class to get the core set of classes. Draw the analysis diagram including relationships and multiplicities.
2. Perform top-level design using CRC cards completing responsibilities for knowing, doing and collaborators for the core set of classes identified in 1.

Steps 1 & 2 can be done in groups. There will be an exercise on analysis & top-level design during the tutorial on Mar. 20. Steps 3 through 6 ***must*** *be individual work.*

3. Perform detailed design to develop a complete (and compilable) set of Java interfaces for the core classes.
4. Perform coding of implementations for the core classes. This may result in discovery of additional support classes which will need to be implemented
5. Implement main classes for three applications:
   a. Initialization of the store (i.e. establishing customer and item lists).
   b. Customer placing an order
   c. Clerk shipping order(s).
6. Perform testing of the system.
7. For production (submission), run the system using the supplied data files.

Suggestions

- All files should be part of a single package and included in the same sub-directory (say `Spartan`) of the `Assign5` directory.
- You can make use of the `Collections` library as discussed in class. The `Collections.jar` file is included in `A5Files.zip`. Unzip it into your `Assign5` directory and add it to the extra class path in the projects for each of the applications (i.e. create, order and ship).
- At implementation, consider using separate classes for GUIs and reports.
- You will likely end up with between 10 and 15 classes and interfaces including classes identified during design, main classes and support classes (e.g. GUIs and reports) identified during coding. There should be interfaces for the classes identified during design, but it is not necessary to have interfaces for support classes (although it doesn't hurt).
- Marks will be assigned to both the quality of the design and the quality of the implementation.

Submission:

Details regarding preparation and submission of assignments in COSC 1P03 are found on the COSC 1P03 website at URL: `http://www.cosc.brocku.ca/Offerings/1P03/AssignGuide.pdf`. This document includes a discussion of assignment preparation, programming standards, evaluation criteria and academic conduct (including styles for citation) in addition to the detailed assignment submission process copied below. **Part of the marks for the assignment will be awarded for programming standards.**

To prepare and submit the assignment electronically from the lab, follow the procedure below:

1. Ensure your folder (say `Assign5`) for the assignment is stored on your `Z:` drive.
2. Ensure that `Collections.jar` is included in the `Assign5` folder and correctly referenced in the three projects in the subfolder. The applications should execute from within the subfolder of the `Assign5` folder.
3. Print (to `PDFCreator`) the `.java` file of all the classes and interfaces for your assignment using the name *`ClassName`*`.pdf` for the class/interface *`ClassName`* and save the `.pdf` file at the **top level** of the project folder (i.e. directly within `Assign5`).
4. Unzip the file `StoreData.txt` from `A5Files.zip`.
5. Run your create application to initialize the store using the file `StoreData.txt` file. This file starts with the number of items in the catalogue and then, for each item, has a line containing the item number, unit price and description. Then the number of customers followed by one line per customer giving the customer number, name, address and password.
6. Unzip `Script.pdf` from `A5Files.zip` and follow instructions, running your order and ship applications as specified. When instructed to save a form, use `File/Save Image of Window`... and save it with the specified name at the **top level** (i.e. in the `Assign5` folder). When instructed to print a report, print to `PDFCreator` with the specified name at the **top level** (i.e. in the `Assign5` folder).
7. Run PuTTY by selecting `PuTTY` under `All Programs` in the `Start` menu.
8. Double-click `sandcastle` in the `Load, save` ... entry.
9. Enter your Brock userid and press the `Enter` key.

10. Once you have the `sandcastle%` prompt, navigate to your project directory for your assignment (say `Assign5`).
    Here are a few useful commands (press `Enter` after typing the command):

`ls –l`	- list files in current directory
`cd <directory name>`	- changes to the specified subdirectory (note, do not include the <>) e.g. `cd Assign5`
`cd ..`	- go up 1 directory level

    Note: If your file or folder names include spaces or special characters, you have to enclose the name in quotes, e.g. `cd "COSC 1P03"`.

11. Once you have confirmed you are in the correct project directory, type the command `submit1p03` and follow the instructions. It is important to note that the script will copy everything from the current directory and its subfolders to the 1P03 electronic drop box. It is important you are in the correct directory when you run the script. The script will confirm what you have submitted.

12. Log off sandcastle by typing `logout`.

For help in submitting an assignment from home see the COSC Help Center at URL: `http://www.cosc.brocku.ca/help/esubmit`.

DrJava

The folder from which you do the electronic submission should contain the project folder, including all files (including those supplied) relevant to the project—the `.java` and `.class` files for the assignment and `.pdf` files for program listings and output.

Other platforms

If you are using an IDE other than DrJava to prepare your assignment at home, you must copy your code into DrJava to create new project(s) and then compile and run and prepare the submission as above. Your electronic submission must only include DrJava project folders and the `.pdf` files as described.