

COMP 2150 Assignment 4 – Summer 2022 Packages

This assignment is due by **August 11th @ 11:59 PM**.

Submission format: On the course page on UMLearn, go to Assessments → Assignments and upload your submissions to the Assignment 4 folder.

- Please follow both the "Programming Standards" and "Assignment Guidelines" for all work you submit.
- Ensure that you submit a signed Blanket Honesty Declaration before the assignment due date.

(Java): Point of Sale: Part 3

For this question you will add additional features to your Point of Sale server from the previous assignment. You may use your own solution, or the posted model solution. The new features will be:

1. A client. Rather than developing this yourself, though, you will be using a pre-compiled Java class given to you, and connecting it to your server.
2. Packages. The client and server will be organized into separate packages, along with a main program that will launch both.
3. A singleton. This will ensure that there is only ever a single instance of your server, no matter how many clients are connected to it.
4. Order cancellation. A transaction (any of the three types) can be cancelled.

Your client will be interactive, implemented using the given InteractiveClient class. This class has only one useful public method:

Method Summery
public static String <code>comp2150.pos.client.nextCommand()</code> Get the next command entered, if any. Returns null if the user chooses to quit.

To use this class, call the `nextCommand()` method repeatedly. It will return strings that you can then translate into commands that you can pass to your server. Use features of your server in your responses to commands; for example, print the total count and quantity of items and cost of a transaction as you add items to it. Error messages from your server can also be printed.

To organize your system, divide it into three packages:

- `comp2150.pos.client` contains the client
- `comp2150.pos.server` will contain your server implementation
- `comp2150.pos.main` contains the main program that starts up the server and the client

Since there will only ever be one server in your system, you can enforce this uniqueness using the Singleton design pattern. Implement the singleton as a class called `A4Server` in the server implementation package. To enforce the information hiding for your server implementation, your server class must be private to the `comp2150.pos.server` package (that is, it should have default or package visibility, and not public).

Any transaction can be cancelled using the new server interface method **`cancelTransaction()`**. Cancelling a purchase will return all the items in the purchase to the inventory; that is, the in-stock quantity will be increased by the amounts from the transaction. Cancelling a backorder will remove backordered items from the inventory backorder quantity and return any purchased items to inventory. Cancelling a return will have no effect on inventory. Cancelled transactions are marked as "cancelled" rather than completed.

There is one other change required by cancellations. Transactions can only be cancelled from the client that created the transaction. Therefore, transactions must keep track of which client created them using a client ID. This ID has been added to **`createTransaction()`**, and is also used in **`cancelTransaction()`**.

These changes require an updated server interface `A4POSServer.java`. Use the inventory file from the previous assignment. With your assignment submission, include an `OUTPUT.txt` file with your assignment submission that shows the output of running your program (copy and paste). Ensure that it includes:

- more than one client;
- purchases, backorders, and returns;
- cancellations of all three types of orders;
- searches (with multiple matches) and summaries; and
- both successful and failed actions (e.g. cancelling from the wrong client).