

Final Implementation for Project 1.

Complete by October 25 (Friday). One per group.

This is to be done on Github, one per group. The assignment requires you to create a final version of the program. Start with the code developed for the Stage 1 implementation and add the necessary code for the features designed in Stage 2.

- 1. Simple version of warehouse (75 points, no waitlists).** In the simple version, we don't have waitlists. If a product runs short or is out of stock when an order is placed, the unfulfilled part of the order is put back on the wishlist. When a shipment of a product is received, the quantity of product that arrives is added to the stock for the product. Orders that have not been filled will have to be individually processed again, when needed.
- 2. Full version of warehouse (150 points).** In the full version, all the features are implemented. If a product runs short or is out of stock when an order is placed, the unfulfilled part of the order is added to the waitlist for the product. When a shipment of a product is received, the waitlist items are filled first. For each waitlist item filled, an invoice is generated for the associated customer. The remaining quantity of product is added to the stock for the product.

How to demonstrate testing? Start a script session in your folder (command "script warehouseTest"). Compile the program (command "javac *.java") and run (command "java UserInterface"). Perform the following sequence of operations. Each operation should be done through a menu option provided by the UI. Note that we are doing queries in between to see if results are as expected.

Create five clients: C1 through C5

Print/display all clients (should show the credit/debit balance for each)

Create five products: P1 through P5, with quantities 10, 20, 30, 40, 50 respectively and unit prices \$1, \$2, \$3, \$4, \$5 respectively.

Print/display all products (should show price and qty for each)

Add to C1's wishlist: 5 each of P1, P3 and P5

Print C1's wishlist

Add to C2's wishlist: 7 each of P1, P2 and P4

Print C2's wishlist

Add to C3's wishlist: 6 each of P1, P2 and P5

Print C3's wishlist

Place order for C2, buy everything (if available) in wishlist

Print/display all clients (should show the credit/debit balance for each)

Place order for C3, buy everything (if available) in wishlist

Print/display all clients (should show the credit/debit balance for each)

Print C2's wishlist

Print C3's wishlist

Print P1's waitlist (for full version of program)

Print P2's waitlist (for full version of program)

Place order for C1, buy everything (if available) in wishlist
 Print/display all clients (should show the credit/debit balance for each)
 Print C1's wishlist
 Record receipt \$100 payment for C1 and C2.
 Print/display all clients (should show the credit/debit balance for each)
 Receive a shipment of 100 items of P1.
 Print/display all products (should show price and qty for each)
 Print/display all clients (should show the credit/debit balance for each)
 Print all invoices for C1
 Print all invoices for C2.

Final group report in D2L.

The final group report must contain the following items:

- 1. Work sharing percentage.** Work-sharing will be done as follows: each group will submit the percentage of work done by each group member, with a brief justification of why that percentage is appropriate. (e.g., Member1 contributed 33% because they attended ..., , completed the following tasks, etc.). If all members are sharing equally, no need for writing a justification. The points adjustment will be done (by me, later on) as follows: *if everyone contributed equally, each member gets 100 points. If a member contributed only half of what they were expected to (e.g., 16% in a group of 3, instead of 33%), 50 of their points will be reassigned to others who did more than 33%.*
- 2. Where to find code.** Name and_starid of group member who's Github assignment has the code, and instructions to compile and run the code.
- 3. Table to report on test results.** Create a table in the following format to report on the tests. The list of tests is given above; results of all the tests should be shown in table.

Version: Full (or Simple)		
Operation Performed	Expected output	Actual output
Create five clients: C1 through C5	Confirmation of creation	Ok (report what you get)
Print/display all clients	C1 <details of C1> Balance:\$0.0 C2 <details of C2> Balance:\$0.0 C3 <details of C3> Balance:\$0.0 C4 <details of C4> Balance:\$0.0 C5 <details of C5> Balance:\$0.0	Report what output you got
Add prod P1, qty 10, price \$1, Add prod P2, qty 20, price \$2 Etc,	Confirmation of creation	