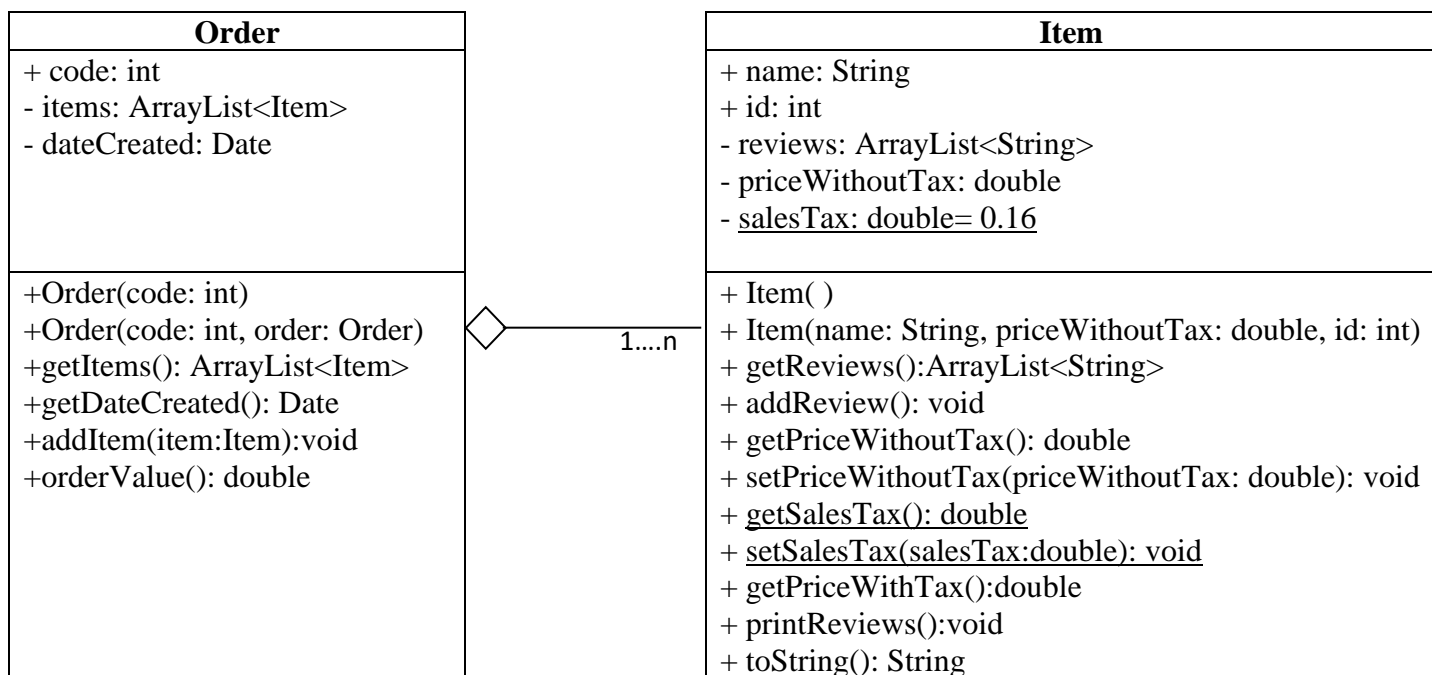




The University of Jordan
Faculty of Engineering and Technology
Department of Computer Engineering

Programming of Networks Protocol
Course Final Project
Eng. Asma Abdel Karim

In this course project, you are required to design and implement an ordering system using a client/server model and the **Java API for TCP**. The basic classes that form orders are shown in the following UML diagram and their implementations are provided in the attached files: Order.java and Item.java.



- You are free to add and modify methods in the Order and Item classes as required.
- You must complete the design by adding classes as needed to implement the requirements below.

The ordering system consists of a server program that maintains a list of items and allows users to login and use the system by processing their requests. There are two types of users (clients): admins who can add, modify, and remove items in the system, and customers who make orders. Note that the client program provides interface to the users through which: 1) they enter requests and data that will be sent to the server and 2) info received from the server are displayed.

1. The server side:

The server must be able to handle several clients at the same time. It must also save info of all items in the ordering system and info of registered clients (even when the server application terminates).

- The server maintains items of the ordering system in file named items.out. This file is already attached with the project files and contains a serialized ArrayList of items. The server must load the ArrayList of items when it is first started. It must update the ArrayList serialized on the file each time an admin makes a change to the ArrayList of items.
- The server maintains info of all users registered to the ordering system. Each client (user) that connects to the server has a user name and a password that he must enter to log in.
- As mentioned earlier, users can be admins or customers. Both must first register (sign up) to the system to have accounts.
 - In order for admins to sign up, they must first enter a code (generated randomly by the program, assume it is sent to them by email/sms). Then if the entered code matches the one generated by the program, they proceed to entering their username and password.
 - In order for customers to sign up, they must register with their username and password.
 - When a user is signing up, the server must make sure that username is not already taken. If so, it must return an error to the client indicating that, and ask him to enter a username and password again.
- A user, whether admin or customer, must log in first to be able to perform operations. The client must specify whether he wants to login as an admin or as a customer. The server must receive the username and the password and make sure that there is a user (admin or customer) with the received name, if not it must return an error to the client indicating that there is no user with the entered name. If the username exists, it must match the received password with the password saved for that user. If they do not match, it must return an error message to the client indicating that. Whenever an error occurs, the login process must be performed all over again.
- When a user is logged in, he can start performing operations based on whether he is an admin or a customer:
 - Admins can perform the following operations on the server: add items to the list of items, remove items from the list, and modify info of items in the list.
 - Customers can make orders by adding new items and list their previous orders.

2. The client side:

- When a client first connects to the server, he will have two options either to sign up or log in.
 - If the client chooses to sign up, he must choose whether to sign up as an admin or a customer. Based on his selection the sign-up process continues.
 - If the client chooses to login, he must also choose to login as either an admin or customer. Then he must enter his username and password. Based on the server response either login succeeds, and he can start performing operations or login fails, and he must enter his username and password again.

- If a client is logged in as an admin, he will have a menu of operations that he can perform which include add item, remove item, modify item. The admin choice must be sent to the server. Based on his choice the program must proceed to take inputs from the user to add, remove, or modify items data (name, id , and price without tax).
- If a client is logged in as a customer, he can perform one of three operations: make new order, list orders, or exit. The customer choice must be sent to the server, and based on his choice the communication proceeds.
 - If the customer chooses to make a new order, list of items in the ordering system must be displayed to him and he can start picking which items to add to the order. After adding each item, the program must ask him whether he wants to add more items or finish the order. When the order is finished, the list of orders including all added items details must be printed with the order value at the end. And the order must be added to the customer orders.
 - If the customer chooses to list his previous orders, all his previous orders codes and dates must be displayed. The user can then perform one of three operations:
 - Reorder: the user must be asked to enter the code of the order he wants to reorder. A new order must be initiated and initialized with the previous order. The program must ask the user whether her want to modify the order by adding items or removing items or finish it. If the user selects to add items, list of items in the ordering system must be displayed to him and he can start picking which items to add to the order. If the user selects to remove items, he must enter which item to remove, and the item must be removed from the order. When the order is finished, it must be displayed.
 - Show order: the user can select one of the orders to show. If this option is selected, the order details including all items in the order and the order value must be printed. The program must then ask the user whether he wants to add a review to one of the items. If so, he must select the item to revies, and enter his review which must be added to the item's reviews. After adding his review, the program must ask him whether he wants to add a review to another item, and so on until the user does not want to add any reviews.
 - After making a new order or listing previous orders, the program must display the main menu of: make new order, list orders, or exit.
 - If the user selects to exit the program, the connection to the server must be terminated, and the client code must be terminated.

Note that your program must make sure that access to the following collections is synchronized:

- Reviews of items: you must make sure that no two or more customers will add to the reviews of an item at the same time.
- Items in the ordering system: you must make sure that when items are being updated by an admin, it cannot be accessed by customers until they are fully updated.

Your final submission must include the following:

1. Your code fully documented.
2. A report that includes the UML class diagram showing the design of your application data and methods.