

# Code Test

## The Auction House

Write an Auction House / Item Store simulator in C++. There should be one server application that accepts connections from multiple clients at the same time over the network and displays a text-based store interface to each.

You may use any language features or standard library functions supported by Visual Studio 2019. If you use any other libraries or frameworks, the build and install process should be automated and no manual setup should be required other than pressing build. Be prepared to motivate your library choices. File based database libraries as well as network libraries are of course allowed.

The application should be self-contained; no external servers or services should be required to run the project.

### Required functionality

- Ability to 'log in'/choose your account. If the account does not exist, it should be created.
- Ability to deposit funds and items. Funds and items are added to your current account. In this simulation, you can freely deposit any amount of funds or items, they will simply appear in your account.
- Ability to withdraw funds and items. In this simulation, withdrawn funds/items are simply removed on the server.
- Ability to put items from your account up for sale. Items must have been previously deposited. A selling fee should be deducted from the seller's account.
- Ability to buy an item that is for sale by someone else. When bought, the item will be deposited into the buyer's account, and the purchase price will be deposited into the seller's account.
- Display sales listings, your inventory, and your funds balance.
- If an item is for sale for more than 5 minutes, it should expire and be returned to the seller. The fee is not refunded.

### Scope

- Items can be free form text; a complete item system is not required.
- No account security is required. No passwords etc.
- You do not need to create a client; you can use an existing telnet client to connect.
- You do not need to implement any of the telnet protocol, raw sockets are OK.

## Bonus functionality

- Notify seller in real-time of a completed sale (if they are currently connected)
- Auction functionality, where a buyer can bid on an item up until the expiry time, at which point the item is “bought” and deposited into the highest bidder’s account.
- Transaction logs for sales, deposits and withdrawals.

## Things to consider

- System correctness. The system must never end up in an inconsistent state and must disallow invalid/incorrect transactions.
- Safety. The system must never lose any funds or items. Think about crashes and error handling.
- Code clarity.
- Performance. Especially while using multiple clients at the same time.

## Delivery

The delivery should consist of a zip file with:

- Application source code:  
You may use either a Visual Studio solution that we can open with VS2019, or a CMake project.  
The project should build without any errors or warnings on windows. Bonus points if it also does so under Clang 11 on Linux.
- Very short documentation:  
General overview how the problem was solved.  
Any problems you encountered and how they were solved.

Please do not include any executable files (BAT, EXE, DLL etc); the mail filter will reject the mail.