# Distributed Systems Assignment

## Name and Student Number

Paul Good C12397836

## Subject Title

Distributed Systems

## Assignment Title

Auction Assignment

## Date

16/11/2016

## Declaration

I declare that this work, which is submitted as part of my coursework, is entirely my own, except where clearly and explicitly stated.

## Command Line Example

### Client

java AuctionClient 127.0.0.1 1234

### Server

java AuctionServer 1234

## Architecture and Implementation

I used our work from lab week 4 task 8 as a skeleton architecture in which there was a server class, a server thread class, a client class and a client thread class.

This allows a runnable server to set up server threads to handle socket communication, read, writes and setup and closing. This was similar for a client, as a client class was also runnable and set up client threads to handle its socket communication, read, writes and setup and closing.

### Server

My auction server has a constructor used to setup the socket and run the thread. Run awaited connecting cleints.

In the addThread method I would broadcast details of the current auction such as the current item, and its current bid price or its reserve price if no one had bided yet. It also sent the time left for this current items auction. Upon timeout the next item would be presented.

I had a method called runtimer which created a TimerTask to run and handle when the next auction would start, manage the auctions arraylist, notify the user that the current auction was finished and who won the item and at which price. It also handles when the entire auction is finished i.e. when all items are sold.

FinishAuction would be called when all items in the auction are sold. It would notify users of what items they had bought and would then handle closing the server and notifying the clients to close also.

The program would take in client requests using the broadcast method and react accordingly. E.g. If a user bids higher than the current highest bid the timer for this items auction would be reset and all other clients connected would be notified of the new bid and who set that bid.

### Client

My auction client has a constructor to set up its socket and run its thread.

Run awaits a user’s input. It ensures this input is higher than the highest bid it has received from the server on connection. If the checks are fine, it writes this bid to the server.

In the handle method the client receives input from the server and reacts to these accordingly. If it receives .bye from the server it will know that the auction has finished and to end its thread and close the program.

If the input from the server contains the string bid, it knows to parse the bid from this input and set it as the new highest bid. The server ensures this is the case in its broadcast method.

Then the server prints the message to the client to show the message from the server containing key information about the state of the auction.

**Extra functionality**: The user is notified after every auction whether he/she has won or which user has won and at which price. Also at the end of all the auctions each user is notified which items they had won throughout the auction and at what price. The server and clients then gracefully shutdown.

## Class Diagram

