For these assignments, you will practice with client-server TCP socket programming in Java (Assignment #2) and C (Assignment #3). Each client must be able to communicate with each server. For each assignment, you will complete both a client and a server.

You will implement an auction protocol for a single item. The server will be able to support multiple simultaneous bidders; thus, it needs to be multithreaded. You must determine where synchronization is needed.

A client will join an auction by sending a JOIN message. A sample JOIN message is as follows:

JOIN eva

The server will respond with an OK message if there is no problem or an ERR message otherwise. A sample OK message is as follows:

OK eva

A sample ERR message is as follows:

ERR dup // if someone else with the same name has already joined
ERR end // if the auction has ended

A client will bid by sending a BID message. A client cannot bid without joining the auction first. A sample BID message is as follows:

BID eva 500

The amount to be bid is an integer. The server will respond with an OK message if the bid is accepted (high enough), a NO message if the bid is too low, or an ERR message otherwise.

A sample ERR message is as follows:

ERR join // if the client has not joined the auction
ERR end // if the auction has ended

The client will determine its status in the auction by sending a WIN? message. A sample WIN? message is as follows:
WIN? eva

The server will respond with a YES message if the client is winning, a NO message if the client is losing, and an ERR message otherwise (i.e., if the client has not joined the auction, or if the auction has ended).

The client may stop bidding by sending a LEAVE message. A sample LEAVE message is as follows:

LEAVE eva

The server will respond with an OK message (as above) if there is no problem or an ERR message otherwise. Three ERR messages are possible:

ERR join // if the client has not joined the auction
ERR win // if the client is winning the auction
ERR end // if the auction has ended

The client may quit communicating with the server by sending a QUIT message. A sample QUIT message is as follows:

QUIT eva

The server will respond with an OK message (as above) if there is no problem or an ERR message otherwise. If the client is winning, the client may not quit. A proper QUIT command will sever the connection between the client and the server.

The client may end the auction by sending an END message. A sample END message is as follows:

END bob

The server will respond with an OK message (as above) if there is no problem or an ERR message otherwise. A sample ERR message is as follows:

ERR active // if the client is active in the auction