Computer Programming Name:________________________

Chapter 4, Exercises 4.46 to 4.52 (use the Auction project)

Assignment 9

Create an auction with a few lots, persons, and bids.

Then use the object inspector to investigate the object structure.

Start with the auction object, and continue by inspecting any further object references you encounter in the objects' fields.

This will be important to understanding the following exercises:

Exercise 4.47 The makeABid method includes the following two statements:

```
Bid bid = new Bid(bidder, value);
boolean successful = selectedLot.bidFor(bid);
```

The bid variable is only used here as a placeholder for the newly created Bid object before it is passed immediately to the lot's bidFor method. In other words, "bid" is only used in the second line you see above. Rewrite these statements to eliminate the "bid" variable by using an anonymous object as seen in the enterLot method.

Exercise 4.48 Add a close method to the Auction class. This should iterate over the collection of lots and print out details of all the lots. Use a for-each loop.

Any lot that has had at least one bid for it is considered to be sold, so what you are looking for is Lot objects whose highestBid field is not null.

Use a local variable inside the loop to store the value returned from calls to the getHighestBid method, and then test that variable for the null value.

For lots with a bidder, the details should include the name of the successful bidder and the value of the winning bid.

For lots with no bidder, print a message that indicates this.

Exercise 4.50 Add a removeLot method to the Auction class, having the following header:

```
/**
* Remove the lot with the given lot number.
* @param number The number of the lot to be removed.
* @return The Lot with the given number, or null if
* there is no such lot.
*/
public Lot removeLot(int number)
```

This method should not assume that a lot with a given number is stored at any particular location within the collection. Remember you may need to use a for-next loop that runs through the ArrayList backwards.

Exercise 4.51 Rewrite getLot so that it does not rely on a lot with a particular number being stored at index (number-1) in the collection.

Search through each lot in lots until you find a match. If no match is found, print out an error message.