Assignment 5 Reflection:

All of the requirements for Assignment 5 are complete. Most of the code for the positive integers program was types from scratch. A majority of the code for the auction program was used from the chapter 3 exercises, and modified to fit the Auction.java.

In the positive integers program, I used the Random function to generate 150 random numbers. I stored both even and odd numbers in a single array, but stored the even numbers starting at the beginning of the array, and the odd numbers starting at the end of the array.

For the auction program, I used the programs that I typed during the chapter 3 explaining how the StackedLists worked. Because of this, I really only had to develop 2 classes, Bidder and Auction. The bidder would store information on a bidder who took position in the auction. If a bidder bid below the max bid of the previous bidder, that bidder was not processed with the bidder class. The Auction class was the demo for the overall program.

DoubleIntegerStackDemo.java

```
There are 68 evens and 82 odds.

68 Even numbers
--------------------------------------------------------------------------------
326     746     286     368     888     796     470     702     740     470
692     452     442     570     666     548     500     268     734     618
34      778     428     716     488     382     334     438     322     762
626     132     672     366     550     536     30      834     654     966
678     910     172     362     960     960     246     268     596     310
726     958     784     926     314     142     522     632     878     548
474     710     524     670     150     368     820     116

82 Odd numbers
--------------------------------------------------------------------------------
875     73      171     583     23      869     305     705     605     481
33      479     363     639     169     341     371     483     219     517
125     821     767     367     339     449     367     299     869     939
711     751     689     935     417     705     311     747     933     493
889     577     747     683     73      3       271     343     293     543
937     645     213     709     397     781     483     867     391     577
257     651     395     555     557     395     971     5       305     577
607     767     489     347     683     821     559     413     631     9
917     11

**** Pop 1 even and Pop 2 Odds ****
67 Even numbers
--------------------------------------------------------------------------------
326     746     286     368     888     796     470     702     740     470
692     452     442     570     666     548     500     268     734     618
34      778     428     716     488     382     334     438     322     762
626     132     672     366     550     536     30      834     654     966
678     910     172     362     960     960     246     268     596     310
726     958     784     926     314     142     522     632     878     548
474     710     524     670     150     368     820

80 Odd numbers
--------------------------------------------------------------------------------
171     583     23      869     305     705     605     481     33      479
363     639     169     341     371     483     219     517     125     821
767     367     339     449     367     299     869     939     711     751
689     935     417     705     311     747     933     493     889     577
747     683     73      3       271     343     293     543     937     645
213     709     397     781     483     867     391     577     257     651
395     555     557     395     971     5       305     577     607     767
489     347     683     821     559     413     631     9       917     11


**** Getters ****
Even Head:      116
Odd Head:       73
```

Auction.java

```
* * *Welcome to the Auction!* * *

----------------------------------------
Please enter bidder's name: John
Enter a bid: 7

New Bid         Result                High Bidder     High Bid        Maximum Bid
7 John          New High Bidder       John            1               7

Enter a new bidder(y/n)? y

----------------------------------------
Please enter bidder's name: Hank
Enter a bid: 5

New Bid         Result                High Bidder     High Bid        Maximum Bid
7 John          New High Bidder       John            1               7
5 Hank          High Bid Inc.         John            5               7

Enter a new bidder(y/n)? y

----------------------------------------
Please enter bidder's name: Jill
Enter a bid: 10

New Bid         Result                High Bidder     High Bid        Maximum Bid
7 John          New High Bidder       John            1               7
5 Hank          High Bid Inc.         John            5               7
10 Jill         New High Bidder       Jill            8               10

Enter a new bidder(y/n)? y

----------------------------------------
Please enter bidder's name: Thad
Enter a bid: 8

New Bid         Result                High Bidder     High Bid        Maximum Bid
7 John          New High Bidder       John            1               7
5 Hank          High Bid Inc.         John            5               7
10 Jill         New High Bidder       Jill            8               10
8 Thad          No Change             Jill            8               10

Enter a new bidder(y/n)? y

----------------------------------------
Please enter bidder's name: Joey
Enter a bid: 15

New Bid         Result                High Bidder     High Bid        Maximum Bid
7 John          New High Bidder       John            1               7
5 Hank          High Bid Inc.         John            5               7
10 Jill         New High Bidder       Jill            8               10
8 Thad          No Change             Jill            8               10
15 Joey         New High Bidder       Joey            11              15

Enter a new bidder(y/n)? n

***The Auction is Over. Here are the results.The winner is on top.

Bidder  Bid
Joey    11
Jill    8
John    5
John    1

Thank you for using auction.
```

Assignment3Project Junit Testing

Methods to Test	Test Cases	Expected Result	Actual Result
Do{}While	**Test Case 1**	**Add member with bid below current bid and max**	**Member is not added to history**
	Test Case 2	**Add member with bid below max but above current bid**	**Current bid is raised for current max bidder**
	Test Case 3	**Add member with bid above max bid**	**Member is set to highest bidder, current bid is raised**

- The easiest way to check the Positive Integer program was to generate random numbers and check to make sure that the even numbers and odd numbers were sorted properly.

- The Auction program was tested as follows:

    - The main part of the program was a do{ }while that processed one bid and then additional bids as the user chose
        - The first test was to add a member that had a bid below the max bid of the previous user, and also below the current bid
            - The result: The member was printed to the summary of bids, but was not printed in the history, as they did not affect the actual bidding process
        - The second test was to add a member that bid above the current bid, but not above the max bid
            - The result: The member was printed to summary of bids, and the current high bidder had their current bid upped to the new the bid
        - The last test was to add a member that bid above the max
            - The result: The member was added to the history of bids, and was also updated to the highest bidder