Discussion S1

- Research the Internet and share an "article" (article, tutorial, or video) of your choice that is relevant to this week's course topics
    -
    - In several paragraphs, discuss your learning experience from reading the article (or watching the video)
    - **Include its URL so that others can check it out**
- Write Java code and create an **ADT Bag** by "implements" the **BagInterface** (attached file)
    - Using an array-based implementation
    - Chapter 2 (of our textbook) covers the implementation well with actual Java code
    - Your job is to "put it together" and test your code with a driver, such as the **OnlineShopper** (also included in the attached file)
- Spend no more than 4 hours completing the assignment, and then articulate your lessons learned

To earn credit for this assignment, reply to this message -

-
    - **[2 Points]** Attach your work - Java source code and/or write-up that you have created
    - **[2 Points]** Discuss -
        - Your research effort as specified
        - Your learning experience from completing the activity
            - 1 new "concept" that you have utilized
            - Challenge(s) that you have encountered
            - 2 or more observations that you have gained
            - Other relevant matters that you would like to share
    - **[1 Point]** Ask a probing question (or two) for fellow classmates to consider

Research

The article "*What are the benefits to the programmer of using Abstract Data Types?"* does an excellent job of breaking down ADT's in a simple and easy to understand way. The video provided in the article is also a great resource to solidify your knowledge. By going over source code with and without ADT's, and written out disadvantages and advantages in basic terms, the concept of ADTs becomes very easy to understand.

What I Learned

I solidified my knowledge that ADTs' are primarily designed for the ease of use for the end user. The user of an ADT needs only to know that a set of operations are available for a data type, but does not need to know how they are applied. In this case the example in the article was writing source code for a game of blackjack. was the creation of source code for a blackjack game. In this scenario, the user understands that the cards need to be shuffled to ensure randomness in the game, but they don't necessarily need to witness the shuffling process firsthand in order for the game to proceed smoothly.

Challenges Encountered

One of the challenges I had didn't have anything to do with the project but it was a good refresher and wanted to mention it in case anyone runs into this issue after a long day. I noticed that I had inadvertently placed two projects under the same workshop, and as a result, I encountered difficulty running my project without interference from the other. Switching my workspace resolved this issue quickly. In terms of the project itself, I made use of a concept called dynamic array resizing. This technique was implemented in the doubleCapacity() method of the ArrayBag class. This dynamic resizing ensures that the array can accommodate additional elements as needed, allowing the bag to grow seamlessly (behind the scenes) without being constrained by a fixed size.

Observations Gained

Generics: Used generics (<T>) which allowed for the creation of classes and methods that operate on different types in a type-safe manner.

Encapsulation: The ArrayBag class demonstrates the principle of encapsulation, where data (the bag array and numberOfEntries) and methods (such as add, remove, etc.) are bundled together and hidden from external access.

Additional Shares

- This project allowed me to work on writing clearer comments to improve readability and helped me work on my organization skills while coding.

Probing Questions for Classmates

How do the limitations on creating generic arrays in Java contribute to type safety?