\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Homework: Week 1 Day 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Research the following Java datatypes and explain the following:

1. List: The “List” component presents the user with a scrolling list of text items. The list can be set up so that the user can choose either one item or multiple items.

2. ArrayList: Resizable-array implementation of the List interface. Implements all optional list operations, and permits all elements, including null. In addition to implementing the List interface, this class provides methods to manipulate the size of the array that is used internally to store the list.

3. Difference in List vs ArrayList: ArrayLists are: 1- Dynamic in nature, 2- Maintains an order in which elements are inserted in the ArrayList, 3- Duplicate elements are allowed.

4. HashMap: Hash table-based implementation of the Map interface. This implementation provides all of the optional map operations and permits null values and the null key.

5. HashTable: Maps keys to values. Any non-null object can be used as a key or as a value.

6. Differences in HashMap and HashTable: 1- Hashtable is synchronized, whereas HashMap is not. This makes HashMap better for non-threaded applications, as unsynchronized Objects typically perform better than synchronized ones. 2- Hashtable does not allow null keys or values. HashMap allows one null key and any number of null values. 3- One of HashMap's subclasses is LinkedHashMap, so in the event that you'd want predictable iteration order (which is insertion order by default), you could easily swap out the HashMap for a LinkedHashMap. This wouldn't be as easy if you were using Hashtable.

7. Set: A collection that contains no duplicate elements. More formally, sets contain no pair of elements e1 and e2 such that e1.equals(e2), and at most one null element. As implied by its name, this interface models the mathematical set abstraction.

8. HashSet: This class implements the Set interface, backed by a hash table (actually a HashMap instance). It makes no guarantees as to the iteration order of the set; in particular, it does not guarantee that the order will remain constant over time. This class permits the null element.

9. ConcurrentHashMap: A hash table supporting full concurrency of retrievals and adjustable expected concurrency for updates.

10. What is hashCode and equals methods and how do the differ: public int hashCode()

Returns a hash code value for the object. public boolean equals(Object obj)

Indicates whether some other object is "equal to" this one.

**PS:** I used Java7 API to get almost all of these definitions.