Discuss **one** the following topics on Sets and Maps for your initial post. Provide a code example to elaborate on the collection you have selected and your thoughts on that collection type.

* Sets
* Maps

Sets in Java are convenient to use for various reasons. Different forms of sets are available, each offering different benefits and abilities. Specifically, Java has three concrete classes: HashSet, LinkedHashSet, and TreeSet. An AbstractSetSets in Java extends from the Collection interface. A set does not allow duplicates, so any that are included will not be stored. The forEach loop can be utilized in a set since the elements are iterable (Liang, 2019/2025). A hash set can be created using a no-arg constructor or through an existing collection (Liang, 2019/2025, sect. 21.2.1). A hash set does not have an order unless a LinkedHashSet is used, which returns the order in the elements were inputted in (Liang, 2019/2025). To organize a set in increasing or decreasing order, utilize the TreeSet class. SortedSets and NavigableSet can also be used through a TreeSet. Using methods like first() or last() will return the first and last elements within a set (Liang, 2019/2025, sect. 21.2.3). If you want to partially return a set, then the headSet(toElement) and tailSet(-fromElement) should be implemented into the code (Liang, 2019/2025). The NavigableSet is an extension of SortedSet and allows the lower(e), floor(e), ceiling(e), and higher(e) to be used to navigate through a set (Liang, 2019/2025, sect. 21.2.3). If you want to remove the last or first element from a tree set, the pollLast() and pollFirst() methods can be used. TreeSet allows us to search them for a specific element using contains (Fadatare, 2024). TreeSets also can remove elements, return the size, and clear the set (Fadatare, 2024).

Here is an example of a Java TreeSet provided by Ramesh Fadatare in 2024 from the Medium website:

import java.util.TreeSet;  
import java.util.Set;  
  
public class TreeSetExample {  
 public static void main(String[] args) {  
 // Creating a TreeSet  
 Set<String> fruits = new TreeSet<>();  
  
 // Adding elements to the TreeSet  
 fruits.add("Apple");  
 fruits.add("Banana");  
 fruits.add("Cherry");  
 fruits.add("Apple"); // Duplicate element  
  
 // Displaying the TreeSet  
 System.out.println("TreeSet: " + fruits);  
 }  
}

**References**

Fadatare, R. (2024, September 20). *Java TreeSet - Ramesh Fadatare - Medium*. Medium. https://rameshfadatare.medium.com/java-treeset-2fe189d44e04

Liang, D. Y. (2025). *Introduction to Java Programming and Data Structures: comprehensive version*. Pearson. (Original work published 2019)

**Assignment Requirements and Grading:**

* 1. An initial post of approximately 250 words is due by **Thursday, 11:59 p.m., CST**.
  2. For the initial post to be considered substantive, it should be at least 250 words in length and fully cover the topics being presented. Single sentence definitions or responses will not be awarded points.
  3. Submit your post by clicking on the assignment link above, then Create Thread. You must create a thread in order to view your peers' posts. Tip: Create your post in a Word document and then copy and paste your work into the thread.
  4. A minimum of three (3) responses, to the original threads of other students, of 100-200 words each are due by **Sunday, 11:59 p.m., CST**.
  5. To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric.](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf)

Hey, Joe! I really enjoyed reading your post for this week! You are right that duplicates are not allowed when using a set. Not remembering this could cause a lot of confusion for those working with a set. I also found this to be a helpful note to remember when completing the program for this module. In a hash set, there is no order, so if you want or need a specific order, a LinkedList, SortedList, or TreeSet are the better options. The example you included perfectly demonstrates how a HashSet can be utilized. Elements in a hashSet can also be removed or inserted.

Hi, Megan! Your discussion post was very nicely said. Sets in Java are truly powerful, and I see myself using them a lot in the future when programming with Java. If any duplicates are entered into a set, they will not be stored. I like how you mentioned that a HashSet does not sort its elements because this can change which set should be utilized. The code you included is a great example of how a HashSet works. There are also options to delete, clear, and insert new elements. There is also a navigableSet and sortedSet that can be used through TreeSet.

Hello, Samir! You did an excellent job on your post for this week. Your explanation for a Map in Java is spot on, and the example you included is a helpful snippet of how it can run. I have not yet used a HashMap in a program, but I imagine it will be handy in the future. The TreeMap is sorted in its natural order, while a LinkedHashMap goes by insertion order, and HashMap does not have an order. I will say that I did feel a little confused about the o(1) and O(logN) complexities. I re-read through those sections in our book but still left a little lost on that part.