

1. ArrayList Operations

- Write a Java program to create an `ArrayList` of integers. Add 5 elements to it, and perform the following operations:
 - Insert an element at a specific position.
 - Remove an element by index.
 - Get an element by index.
 - Iterate through the list using a `for` loop.

2. LinkedList vs ArrayList

- Create two lists, one using `ArrayList` and one using `LinkedList`. Add the same elements to both. Measure and compare the time taken for adding, removing, and searching operations in both lists.

3. HashMap Basics

- Write a program to store and display student names (`String`) and their marks (`Integer`) using a `HashMap`. Perform the following:
 - Add 5 student entries.
 - Update the marks of a student.
 - Remove a student entry.
 - Iterate and print all entries using `entrySet()`.

4. Sorting a List

- Create a `List` of custom objects `Employee` with fields like `id`, `name`, and `salary`. Write a program to:
 - Sort the list by `name` in ascending order.
 - Sort the list by `salary` in descending order using `Comparator`.

5. Set Interface

- Write a program to demonstrate the use of a `HashSet` to store unique student names. Add multiple names to the set and show that duplicate entries are not allowed.
- Use a `TreeSet` to store student names in a sorted order.

6. Queue Interface

- Write a program that simulates a job queue using a `PriorityQueue`. Add several jobs with priorities and print them in the order they are processed (i.e., based on their priority).

7. Custom Sorting in TreeMap

- Create a `TreeMap` where the key is a custom class `Person` (with fields like `age` and `name`), and values are `Strings` representing the address. Ensure that the `TreeMap` is sorted by `age` using a custom `Comparator`.

8. Synchronized Collections

- Create a program that uses `Collections.synchronizedList()` to create a thread-safe version of an `ArrayList`. Implement a multi-threaded program where multiple threads add elements to this synchronized list.

9. Concurrent Collections

- Write a program to demonstrate the use of `ConcurrentHashMap`. Create a concurrent map of employee IDs and names. Simulate multiple threads trying to read and update the map simultaneously.

10. Stream API with Collections

- Create a `List` of integers and perform the following using the Stream API:
 - Filter out all even numbers.
 - Find the maximum number in the list.
 - Sort the numbers in descending order.
 - Print each element using the `forEach` method.