Java Map Interface - Programming Exercises (From Scratch to Advanced)

PART 1: Core Map Interface - Basic Understanding

1. PhoneBook Application:

   - Map a person's name to their phone number.
   - Operations: Add, Delete, Search, List.
   - Use containsKey(), validate input.

2. Student Marks Management:

   - Map<Integer, Double>: roll no -> marks.
   - Add, update, calculate average/highest, display topper.

3. Character Frequency Counter:

   - Map<Character, Integer>
   - Input: "banana" -> Output: {b=1, a=3, n=2}

4. Dictionary Application:

   - Map<String, String>: word -> meaning.
   - Add, search, update, remove. Sorted output with TreeMap.

5. Inventory Management:

   - Map<String, Integer>: item name -> quantity.
   - Add, update, view, remove items.

6. Product Price Mapping:

   - Map<String, Double>: sort by price.

7. Group Students by Class:

   - Map<String, List<String>>: Group names by class.

PART 2: Map Implementations - Concept & Usage

Java Map Interface - Programming Exercises (From Scratch to Advanced)

HashMap:

8. Login System - Map<String, String>: Add, validate, reset passwords.

9. Word Count from File - Map<String, Integer>: Read file, count words.

LinkedHashMap:

10. Browser History - Track visited URLs in order.

11. LRU Cache Simulation - Maintain max 5 entries using removeEldestEntry().

TreeMap:

12. Sorted Dictionary - TreeMap<String, String>: Alphabetical ordering.

13. Custom Sorting - Comparator-based sorting of employees by salary/name.

Hashtable:

14. Secure Login - Demonstrate null-safety and thread-safety.

EnumMap:

15. Weekly Planner - EnumMap<Day, String>: Task per day.

WeakHashMap:

16. GC Demonstration - Store temp objects and observe cleanup.

IdentityHashMap:

17. Identity Comparison - Show difference using new String("Java").

PART 3: Advanced Functional Operations & Nested Maps

18. Group Employees by Department:

- Map<String, List<Employee>>: Grouping & highest salary.

19. Nested Map - Class Performance Tracker:
    - Map<String, Map<String, Integer>>: Add student marks, find topper.

20. ConcurrentHashMap:
    - Multi-thread stock updates.

21. Bank Balance Tracker:
    - Use computeIfAbsent, merge, getOrDefault for transactions.

22. Shopping Cart System:
    - Map<Product, CartItem>: Add, update, remove, calculate total.

23. Student Attendance System:
    - Map<Date, List<Student>>: Track attendance per day.

PART 4: Interview-Theory Questions on Map

1. Differences between HashMap, Hashtable, and ConcurrentHashMap.
2. Internal working of HashMap (hashing, buckets, collision).
3. hashCode() conflicts and chaining.
4. Importance of equals() and hashCode().
5. Load factor, resizing in HashMap.
6. Can TreeMap accept null keys?
7. LinkedHashMap vs HashMap (insertion order).
8. Use case of WeakHashMap (GC cleanup).
9. IdentityHashMap usage (== instead of equals()).
10. Custom class as map key - when to override equals/hashCode.

BONUS CHALLENGES

Java Map Interface - Programming Exercises (From Scratch to Advanced)

24. Topper per subject from nested TreeMap.

25. Track product orders per customer: Map<String, List<Order>>.

26. Convert JSON to Map<String, Object> using Jackson.

27. Convert Map to two Lists (keys and values).

28. Remove duplicates using HashMap.