Assignment Questions on Java Map Interface

1. **Creating and Adding Entries**:

   Write a Java program to create a HashMap of integers and strings. Add five key-value pairs where the keys are integers and the values are strings. Display the contents of the map.

2. **Accessing Values**:

   Write a Java program to create a TreeMap where the keys are strings representing country names and the values are their respective capitals. Add five country-capital pairs to the map. Retrieve and print the capital of a specific country.

3. **Updating Values**:

   Write a Java program to create a LinkedHashMap of String keys and Integer values representing student names and their scores. Add five student-score pairs to the map. Update the score for a specific student and display the updated map.

4. **Removing Entries**:

   Write a Java program to create a HashMap of String keys and Double values representing product names and their prices. Add five product-price pairs to the map. Remove an entry with a specific key and display the remaining entries.

5. **Checking Existence**:

   Write a Java program to create a TreeMap of Integer keys and String values representing employee IDs and their names. Add five employee-ID pairs to the map. Check if a specific ID exists in the map and display the result.

6. **Iterating through a Map**:

   Write a Java program to create a HashMap of String keys and Integer values where the keys are city names and the values are their populations. Add five city-population pairs to the map. Use a for-each loop and an Iterator to iterate through the map and print each key-value pair.

7. **Sorting a Map by Keys**:

   Write a Java program to create a TreeMap of String keys and Double values representing movie titles and their ratings. Add five movie-rating pairs to the map. Display the map sorted by keys (movie titles).

8. **Sorting a Map by Values**:

Write a Java program to create a HashMap of String keys and Double values representing product names and their prices. Add five product-price pairs to the map. Sort the map by values (prices) in ascending order and display the sorted entries.

9. **Merging Two Maps**:

   Write a Java program to create two HashMap objects. The first map should contain five student names and their scores, and the second map should contain additional student names and scores. Merge the second map into the first map and display the merged map.

10. **Finding the Maximum and Minimum Values**:

    Write a Java program to create a TreeMap of String keys and Integer values representing book titles and their ratings. Add five book-rating pairs to the map. Find and display the book with the highest and lowest rating.

11. **Student Grade Management**:

    Write a Java program to create a HashMap of String keys and ArrayList<Integer> values where the keys are student names and the values are lists of their grades. Add grades for five students. Provide functionality to calculate and display the average grade for each student.

12. **Library System**:

    Write a Java program to create a LinkedHashMap of String keys and Book values where the keys are book titles and the values are Book objects. Each Book object should have properties such as author and year. Add five books to the map. Implement functionality to search for a book by its title and display its details.

13. **Employee Salary Management**:

    Write a Java program to create a TreeMap of Integer keys and Employee values where the keys are employee IDs and the values are Employee objects. Each Employee object should have properties such as name and salary. Add five employees to the map. Implement functionality to find and display the employee with the highest salary.

14. **Product Inventory System**:

    Write a Java program to create a HashMap of String keys and Product values where the keys are product codes and the values are Product objects. Each Product object should have properties such as name and quantity. Add five products to the map. Implement functionality to update the quantity of a product and display the updated inventory.

15. **Order Management System**:

Write a Java program to create a TreeMap of String keys and Order values where the keys are order IDs and the values are Order objects. Each Order object should have properties such as customerName and totalAmount. Add five orders to the map. Implement functionality to find and display all orders with amounts greater than a specified value.