Map Problems

- 1. Map with Custom Key and Value Problem:

- Create a Map where the key is an Employee object (with id and name) and the value is a Department object. Populate the map with data and implement a method to retrieve the department of an employee by their ID.

- 2. Frequency Counter Problem:

- Given a list of Book objects (with title and author), create a Map where the key is the author and the value is the count of books written by that author.

- 3. Grouping by Category Problem:

- Create a Map where the key is a Category object (with id and name) and the value is a list of Product objects. Implement a method to fetch all products under a specific category.

- 4. Find Most Expensive Product Problem:

- Given a Map<Product, Double> where the key is a Product object and the value is its price, write a method to find the most expensive product.

- 5. Student Grades Problem:

- Create a Map where the key is a Student object (with id and name) and the value is a List<Integer> representing their grades. Write a method to calculate the average grade for a given student.

- 6. Inverted Map Problem:

- Given a Map<Employee, Department>, write a method to invert the map so that the key is a Department object, and the value is a list of Employee objects working in that department

- 7. Bank Accounts Problem:

- Create a Map where the key is an AccountHolder object (with id, name, and email) and the value is an Account object (with accountNumber and balance). Write a method to deposit an amount into a specific account.

- 8. Map of Maps Problem:

- Create a Map<String, Map<Integer, Product>> where the outer key is a Category name, the inner key is a Product ID, and the value is a Product object. Implement a method to retrieve a product by category and ID.

- 9. Multi-Level Sorting Problem:

- Given a Map<Employee, Double> where the key is an Employee object and the value is their salary, write a method to sort the entries first by salary in descending order and then by employee name in ascending order.

- 10. Attendance Tracker Problem:

- Create a Map<Subject, Map<Student, Boolean>> where the outer key is a Subject object, the inner key is a Student object, and the value is a Boolean indicating attendance. Write a method to count how many students attended a given subject.