Lab: Movie Ratings Manager — HashMap in Java

Objectives:

- Create and use HashMap and nested HashMap structures
- Perform insertion, retrieval, update, and removal operations
- Iterate over keys, values, and entries
- Sort a HashMap by key or value
- Use maps to calculate aggregate information

Scenario

You've been hired to prototype a simple Movie Ratings Manager for a streaming company.
Each user can rate multiple movies. The program should allow storing, updating, and viewing ratings, and calculate average scores per movie.

You'll use Java and focus only on the logic — no GUI is needed.
Use a main class named **MovieRatingsManager**.

Part 1 – Basic Structure

- Create a HashMap to store ratings in this format:
- user → (movie , rating)

Example:

Alice → {Inception=9, Titanic=7}
Bob → {Inception=8, Avatar=9}

- Implement a method to add a rating for a given user and movie.
  - If the user doesn't exist, create a new entry.
  - If the user already rated the movie, update the rating.
  - Print a message when a rating is added or updated.

Part 2 – Display and Search

- Create a method to print all ratings in a readable format.
  Example:

  Alice: Inception=9, Titanic=7
  Bob: Inception=8, Avatar=9

- Implement a method to find all ratings for a given movie.
  - Input: movie name (via Scanner).
  - Output: which users rated it and their scores.

Part 3 – Calculating Averages (going deeper!)

- Write a method to calculate the average rating per movie using your HashMap data.
  - Hint: you'll need a temporary map to group ratings by movie.
- Display the average ratings in the console.

Part 5 – Updating and Removing

- Implement a method to remove a user’s rating for a movie.
  - If the movie doesn’t exist for that user, print a message.
- Allow the user to update a rating interactively (ask for user, movie, new score).