Metropolitan University of Tirana
Data Structures
HashMaps in Java

11.1 Leveraging HashMaps to Manage Company Records

You are tasked with building a data structure to manage employee records for a company using hash maps. The company has multiple departments, and each department can have multiple employees. Use a hash map to store the department names as keys and a list of employees as values. Implement the following operations:

- **Add Department**: Write a method `addDepartment` that takes a department name as input and adds it to the hash map. If the department already exists, display an appropriate message.
- **Add Employee**: Write a method `addEmployee` that takes a department name and an employee name as input and adds the employee to the respective department. If the department doesn't exist, display an appropriate message.
- **Remove Employee**: Write a method `removeEmployee` that takes a department name and an employee name as input and removes the employee from the respective department. If the department or employee doesn't exist, display an appropriate message.
- **Get Employees in Department**: Write a method `getEmployeesInDepartment` that takes a department name as input and returns a list of employees in that department. If the department doesn't exist, display an appropriate message.
- **Get Department for Employee**: Write a method `getDepartmentForEmployee` that takes an employee name as input and returns the department in which the employee works. If the employee doesn't exist, display an appropriate message.
- **Get Employee with Highest Salary**: Write a method `getHighestSalary` that returns the employee with the highest salary across all departments

To implement the above functionalities, make sure to develop the class `Employee` with at least the properties `name` and `salary` and a `main` class where you can test your code

In addition, you can further extend this application by implementing the following (or similar) functionalities

- **Update Employee Details**: Implement a method that allows updating the details of an employee, such as their name or salary. This method should search for the employee in the departments and update their details if found.
- **Get Total Employee Count**: Create a method that calculates and returns the total count of employees across all departments.
- **Get Average Salary per Department**: Implement a method that calculates and displays the average salary for each department. This can involve iterating through the departments, calculating the average salary of the employees in each department, and displaying the results.

- **Sort Employees by Salary**: Add a functionality to sort the employees within each department based on their salary in ascending or descending order. This can help in identifying the highest or lowest paid employees in each department.
- **Search Employees by Name**: Include a method that allows searching for employees by their name. It should return a list of employees that match the given name or a partial name.