# Java Assignments: Access Modifiers & Static Members

## 1. Access Modifiers – Scenario 1: HR Employee Management System

Context:
Your company’s HR software needs to manage sensitive employee data.
- Employee class holds sensitive data like salary (should be private), employee ID (default), and department (protected).
- HRPortal (same package) needs access to non-sensitive info.
- PayrollService (different package) needs access to salary but can’t directly read it—must call a public method.
- Manager (different package, subclass of Employee) needs department info for reports.

Assignment Tasks:
1. Implement Employee with private, default, protected, and public members.
2. Write code in each class to demonstrate what can and cannot be accessed.
3. Document access rules observed from compilation errors.

## 2. Access Modifiers – Scenario 2: Hospital Management System

Context:
In a hospital app:
- Patient class contains sensitive medical history (private), hospital ID (default), insurance details (protected), and name (public).
- Reception (same package) must print patient name and ID but not medical history.
- InsuranceClaim (different package) should access insurance details via inheritance.
- Billing (different package, non-subclass) should only use public getters.

Assignment Tasks:
1. Implement Patient with correct modifiers.
2. Demonstrate that Reception, InsuranceClaim, and Billing follow the security rules automatically via compiler.
3. Create a table mapping each field to which classes can access it.

## 3. Static Data Members & Methods – Scenario 1: E-Commerce Order Tracking

Context:
You are building an order tracking system:
- Order class has:
- Static counter for total orders placed.
- Static method to display total orders.
- Instance fields for orderId, customerName, product.
- Every new order should increment the total order counter.

Assignment Tasks:
1. Create multiple Order objects from Main class.
2. Verify that the counter value is shared among all objects.
3. Demonstrate that you can call the static method without creating an Order object.

## 4. Static Data Members & Methods – Scenario 2: Banking Interest Rate

Context:
Bank interest rate is the same for all customers:
- BankAccount class contains:
- Static field interestRate.
- Static method updateInterestRate().
- Instance fields for accountNumber and balance.
- Changing the interest rate should reflect for all account holders.

Assignment Tasks:
1. Create 3 BankAccount objects with different balances.
2. Update the interest rate via static method and verify it changes for all accounts.
3. Show how the static member behaves even without creating any objects.