Experiment - 9

1. **Bank Account and Transaction Scenario**

   Imagine you're developing a banking application where each bank account needs to keep track of its balance. A **BankAccount** class represents the account, with a private **balance** field and a private method to update this balance. You also want to allow the account to record and manage transactions internally.

   Tasks:

   - Create a **BankAccount** class with a private field **balance** (e.g., double balance) and a private method **updateBalance(double amount)** to adjust the balance by a specific amount.
   - Create an inner class **Transaction** within **BankAccount**, which has a method **processTransaction(double amount)**. This method should modify the **balance** by calling the **updateBalance** method in **BankAccount**.
   - In a separate method in **BankAccount**, such as **performTransaction(double amount)**, create an object of the **Transaction** inner class and call **processTransaction** to adjust the balance. Display the effect on **BankAccount's** balance after the transaction.

2. **Security Access within a Banking System**

   In the same banking application, determine if the **BankAccount** class (the outer class) can access private elements of the **Transaction** inner class, such as a transaction ID or date field, which may be used internally by the **Transaction** class for security or tracking purposes.

   Tasks:

   - Add a private field (e.g., **String transactionId**) and a private method (e.g., **generateTransactionId()**) to the **Transaction** inner class.
   - Demonstrate if the **BankAccount** outer class can directly access these private fields or methods in **Transaction**.