Problem Statement 3:

Design a Java program illustrating the concept of encapsulation. Create a class "BankAccount" with private attributes like balance and methods for deposit and withdrawal, ensuring encapsulation principles are followed.

Solution:

```
class BankAccount{
    private double balance;
    public BankAccount(double initialBalance){
        //Contructor to initialize the balance
        balance=initialBalance;
    }

    public void deposit(double amount){
        if(amount>0){
            balance+=amount;
            System.out.println("Successfully debited "+amount+" to your Account");
        }
        else{
            System.out.println("Invalid amount. DEPOSIT FAILED !!");
        }
    }
    public void withdraw(double amount){
        if(amount>balance){
            System.out.println("Don't Have Enough Balance. WITHDRAW FAILED!!");
        }
        else {
            balance-=amount;
            System.out.println("An Amount of "+amount+" withdrawn Successfully");
        }
    }
    // only Allowing to read the balance
    public double getBalance(){
        return balance;
    }
}
```

```java
class Main {
    public static void main(String[] args) {
     // Create a BankAccount object
     BankAccount myAccount = new BankAccount(10000.0);
         // Display balance
        System.out.println("Current Balance: Rs " + myAccount.getBalance());

        // Deposit money
        myAccount.deposit(500);

         // Display balance
        System.out.println("Current Balance: Rs " + myAccount.getBalance());

        // Withdraw money
        myAccount.withdraw(200);

         // Display balance
        System.out.println("Current Balance: Rs " + myAccount.getBalance());

        // Attempt to withdraw more than balance
        myAccount.withdraw(15000);

        // Display balance
        System.out.println("Current Balance: Rs " + myAccount.getBalance());
    }
}
```