Name:- Prateek choudhary

Roll no:- 023/SCA/BCA(AIML)/052

BANK ACCOUNT SIMULATION

```
import java.util.*;


// Custom Exception

class InsufficientFundsException extends Exception {

   public InsufficientFundsException(String message) {

      super(message);

   }

}


// Base Class

class BankAccount {

   protected String accountNumber;

   protected String name;

   protected double balance;

   protected List<String> transactions = new ArrayList<>();


   public BankAccount(String accountNumber, String name, double initialBalance) {

      this.accountNumber = accountNumber;

      this.name = name;

      this.balance = initialBalance;

   }
```

```
    public void deposit(double amount) throws IllegalArgumentException {
        if (amount <= 0) throw new IllegalArgumentException("Deposit must be positive.");
        balance += amount;
        transactions.add("Deposit: ₹" + amount + " | Balance: ₹" + balance);
    }

    public void withdraw(double amount) throws InsufficientFundsException {
        if (amount <= 0) throw new IllegalArgumentException("Withdrawal must be positive.");
        if (amount > balance) throw new InsufficientFundsException("Insufficient balance.");
        balance -= amount;
        transactions.add("Withdraw: ₹" + amount + " | Balance: ₹" + balance);
    }

    public double getBalance() {
        return balance;
    }

    public void miniStatement() {
        System.out.println("\nMini Statement for " + name + " (Account: " + accountNumber + ")");
        int start = Math.max(0, transactions.size() - 5);
        for (int i = start; i < transactions.size(); i++) {
            System.out.println(transactions.get(i));
        }
        System.out.println("Current Balance: ₹" + balance);
    }
}
```

```
// Derived class: Savings Account
class SavingsAccount extends BankAccount {
    private double interestRate = 0.03;

    public SavingsAccount(String accountNumber, String name, double balance) {
        super(accountNumber, name, balance);
    }

    public void addInterest() {
        double interest = balance * interestRate;
        balance += interest;
        transactions.add("Interest Added: ₹" + interest + " | Balance: ₹" + balance);
    }
}

// Derived class: Current Account
class CurrentAccount extends BankAccount {
    private double overdraftLimit = 1000;

    public CurrentAccount(String accountNumber, String name, double balance) {
        super(accountNumber, name, balance);
    }

    @Override
    public void withdraw(double amount) throws InsufficientFundsException {
        if (amount <= 0) throw new IllegalArgumentException("Withdrawal must be positive.");
        if (amount > balance + overdraftLimit)
            throw new InsufficientFundsException("Exceeds overdraft limit.");
        balance -= amount;
        transactions.add("Withdraw: ₹" + amount + " | Balance: ₹" + balance);
    }
}

// Main Class
public class BankSimulation {
    public static void main(String[] args) {
        try {
            SavingsAccount alice = new SavingsAccount("1001", "Alice", 5000);
            alice.deposit(2000);
            alice.withdraw(1500);
            alice.addInterest();
            alice.miniStatement();

            CurrentAccount bob = new CurrentAccount("1002", "Bob", 3000);
            bob.withdraw(3500); // Uses overdraft
            bob.deposit(1000);
            bob.miniStatement();

        } catch (IllegalArgumentException | InsufficientFundsException e) {
            System.out.println("Transaction Error: " + e.getMessage());
        }
    }
}
```