Practical No. 12

Name: Bhairavi Narendra Rewatkar

Roll No.: DMET1221006

Subject: Blockchain Technology Laboratory

Date: 27/01/2025

Title: Smart Contracts

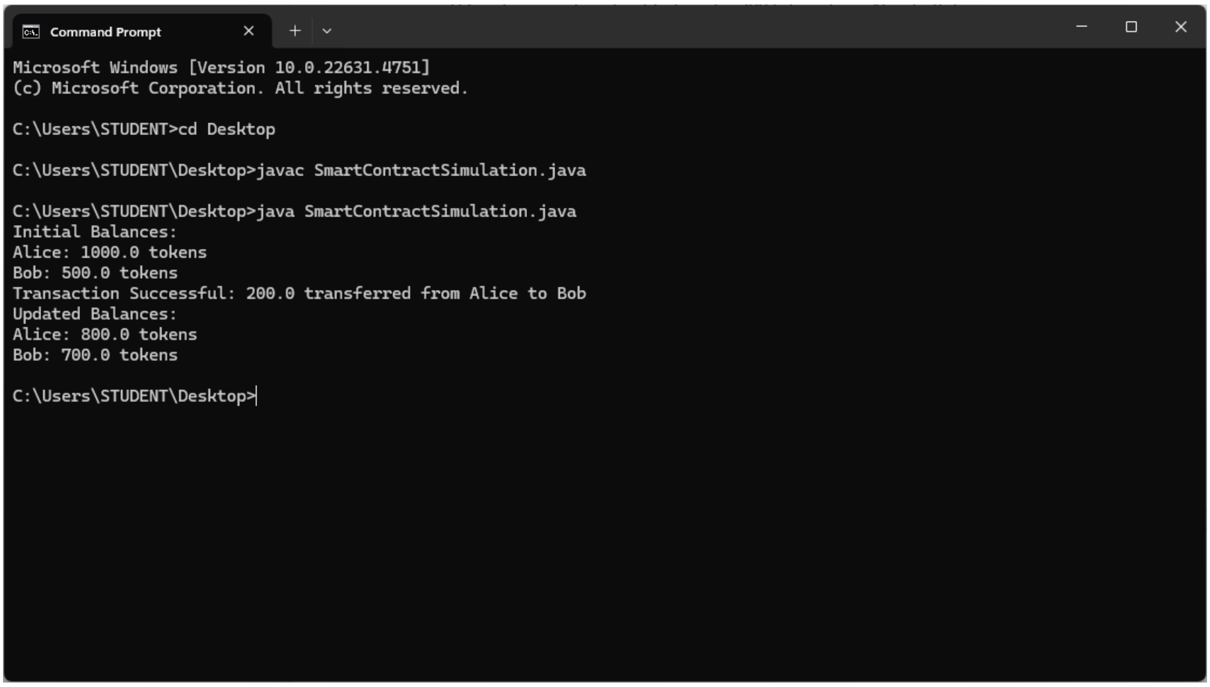
Aim: Write a program to simulate smart contract functionality using Java.

Source Code:

```
public class SmartContractSimulation {
    // Account class representing a user with an account ID and balance
    static class Account {
        private String accountId;
        private double balance;
        // Constructor to initialize account with ID and balance
        public Account(String accountId, double balance) {
            this.accountId = accountId;
            this.balance = balance;}
        // Method to get the current balance
        public double getBalance() {
            return balance;        }
        // Method to credit (increase) balance
        public void credit(double amount) {
            this.balance += amount;        }
        // Method to debit (decrease) balance if sufficient funds are available
        public boolean debit(double amount) {
            if (amount <= this.balance) {
                this.balance -= amount;
                return true;        }
            return false;      }
        // Method to get account ID
        public String getAccountId() {
            return accountId;        }    }
```

```
// SmartContract class simulates the transfer of tokens between two accounts
static class SmartContract {
    private Account sender;
    private Account receiver;
    private double transferAmount;
    // Constructor to initialize the smart contract with sender, receiver, and transfer amount
    public SmartContract(Account sender, Account receiver, double transferAmount) {
        this.sender = sender;
        this.receiver = receiver;
        this.transferAmount = transferAmount;     }
    // Method to execute the smart contract (token transfer)
    public String execute() {
        if (sender.debit(transferAmount)) {
            receiver.credit(transferAmount);
            return "Transaction Successful: " + transferAmount +
                    " transferred from " + sender.getAccountId() + " to " +
                    receiver.getAccountId();
        } else {
            return "Transaction Failed: Insufficient balance in sender's account.";   }}   }
// Main method to simulate smart contract execution
public static void main(String[] args) {
    // Creating two accounts: sender and receiver
    Account sender = new Account("Alice", 1000); // Alice has 1000 tokens
    Account receiver = new Account("Bob", 500);  // Bob has 500 tokens
    // Display initial balances
    System.out.println("Initial Balances:");
    System.out.println("Alice: " + sender.getBalance() + " tokens");
    System.out.println("Bob: " + receiver.getBalance() + " tokens");
    // Create a smart contract to transfer 200 tokens from Alice to Bob
    SmartContract contract = new SmartContract(sender, receiver, 200);
    // Execute the smart contract
    String result = contract.execute();
System.out.println(result);
// Display updated balances after the transaction
System.out.println("Updated Balances:");
System.out.println("Alice: " + sender.getBalance() + " tokens");
System.out.println("Bob: " + receiver.getBalance() + " tokens");}}
```

Output:



```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\STUDENT>cd Desktop

C:\Users\STUDENT\Desktop>javac SmartContractSimulation.java

C:\Users\STUDENT\Desktop>java SmartContractSimulation.java
Initial Balances:
Alice: 1000.0 tokens
Bob: 500.0 tokens
Transaction Successful: 200.0 transferred from Alice to Bob
Updated Balances:
Alice: 800.0 tokens
Bob: 700.0 tokens

C:\Users\STUDENT\Desktop>
```