FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING
Department of Computer Engineering

1. **Course , Subject & Experiment Details**

Academic Year	2022-23	Estimated Time	02 - Hours
Course & Semester	B.E. (CMPN)- Sem VII	Subject Name & Code	BCT - (CSDC7022)
Chapter No.	03	Chapter Title	Programming for Blockchain

Practical No:	3
Title:	Transaction using Solidity
Date of Performance:	22/08/2022
Date of Submission:	29/08/2022
Roll No:	8953
Name of the Student:	Brendan Lucas

Evaluation:

Sr. No	Rubric	Grade
1	On time submission Or completion (2)	
2	Preparedness(2)	
3	Skill (4)	
4	Output (2)	

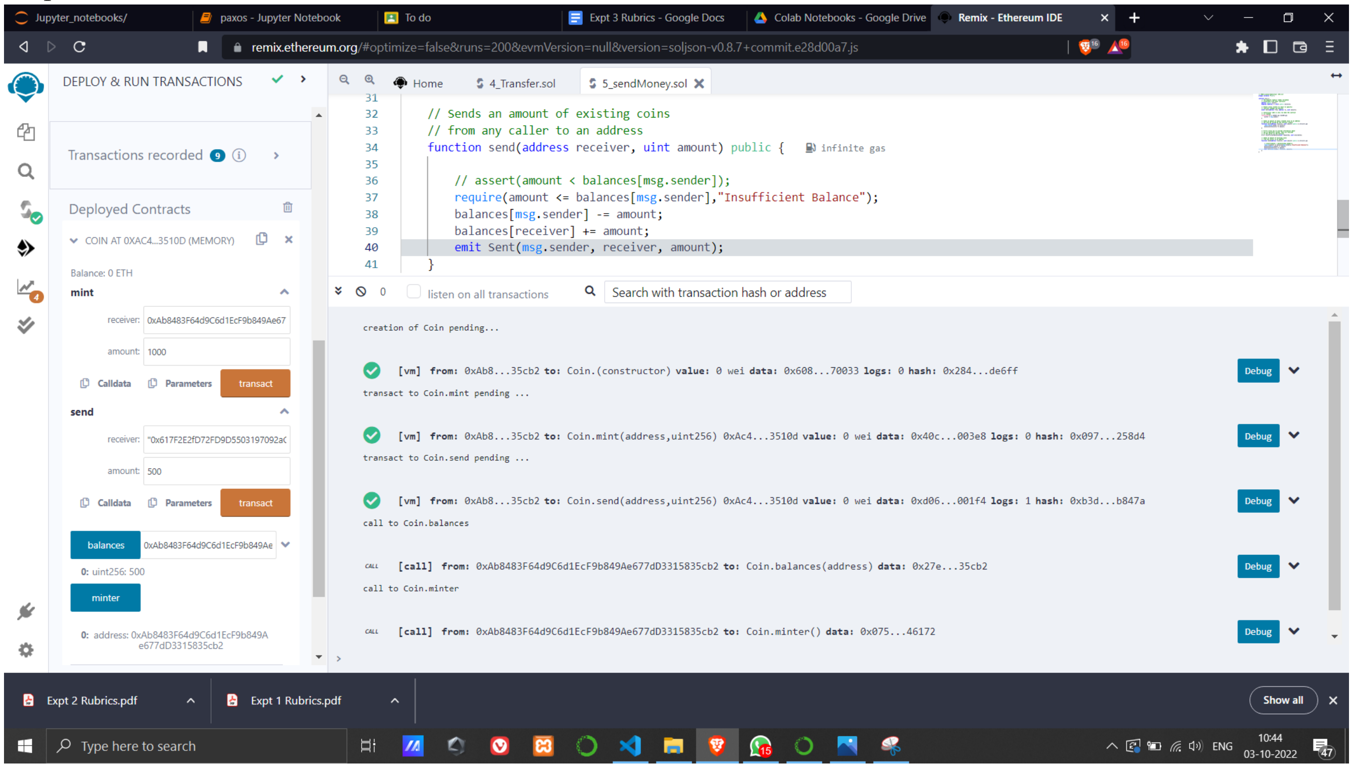
Signature of the Teacher:

Date:

Code:

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.4;

contract Coin {
    // The keyword "public" makes variables
    // accessible from other contracts
    address public minter;
    mapping (address => uint) public balances;

    // Events allow clients to react to specific
    // contract changes you declare
    event Sent(address from, address to, uint amount);

    // Constructor code is only run when the contract
    // is created
    constructor() {
        minter = msg.sender;
    }

    // Sends an amount of newly created coins to an address
    // Can only be called by the contract creator
    function mint(address receiver, uint amount) public {
        require(msg.sender == minter);
        balances[receiver] += amount;
    }

    // Errors allow you to provide information about
    // why an operation failed. They are returned
    // to the caller of the function.
    error InsufficientBalance(uint requested, uint available);

    // Sends an amount of existing coins
    // from any caller to an address
    function send(address receiver, uint amount) public {

        // assert(amount < balances[msg.sender]);
        require(amount <= balances[msg.sender],"Insufficient Balance");
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }
}
```

Output:



DEPLOY & RUN TRANSACTIONS

Transactions recorded 9

Deployed Contracts

COIN AT 0XAC4...3510D (MEMORY)

Balance: 0 ETH

mint
receiver: 0xAb8483F64d9C6d1EcF9b849Ae67
amount: 1000

send
receiver: "0x617F2E2fD72FD9D5503197092a(
amount: 500

balances 0xAb8483F64d9C6d1EcF9b849Ae
0: uint256: 500

minter
0: address: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

```
// Sends an amount of existing coins
// from any caller to an address
function send(address receiver, uint amount) public {

    // assert(amount < balances[msg.sender]);
    require(amount <= balances[msg.sender],"Insufficient Balance");
    balances[msg.sender] -= amount;
    balances[receiver] += amount;
    emit Sent(msg.sender, receiver, amount);
}
```

```
creation of Coin pending...

[vm] from: 0xAb8...35cb2 to: Coin.(constructor) value: 0 wei data: 0x608...70033 logs: 0 hash: 0x284...de6ff
transact to Coin.mint pending ...

[vm] from: 0xAb8...35cb2 to: Coin.mint(address,uint256) 0xAc4...3510d value: 0 wei data: 0x40c...003e8 logs: 0 hash: 0x097...258d4
transact to Coin.send pending ...

[vm] from: 0xAb8...35cb2 to: Coin.send(address,uint256) 0xAc4...3510d value: 0 wei data: 0xd06...001f4 logs: 1 hash: 0xb3d...b847a
call to Coin.balances

[call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: Coin.balances(address) data: 0x27e...35cb2
call to Coin.minter

[call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: Coin.minter() data: 0x075...46172
```