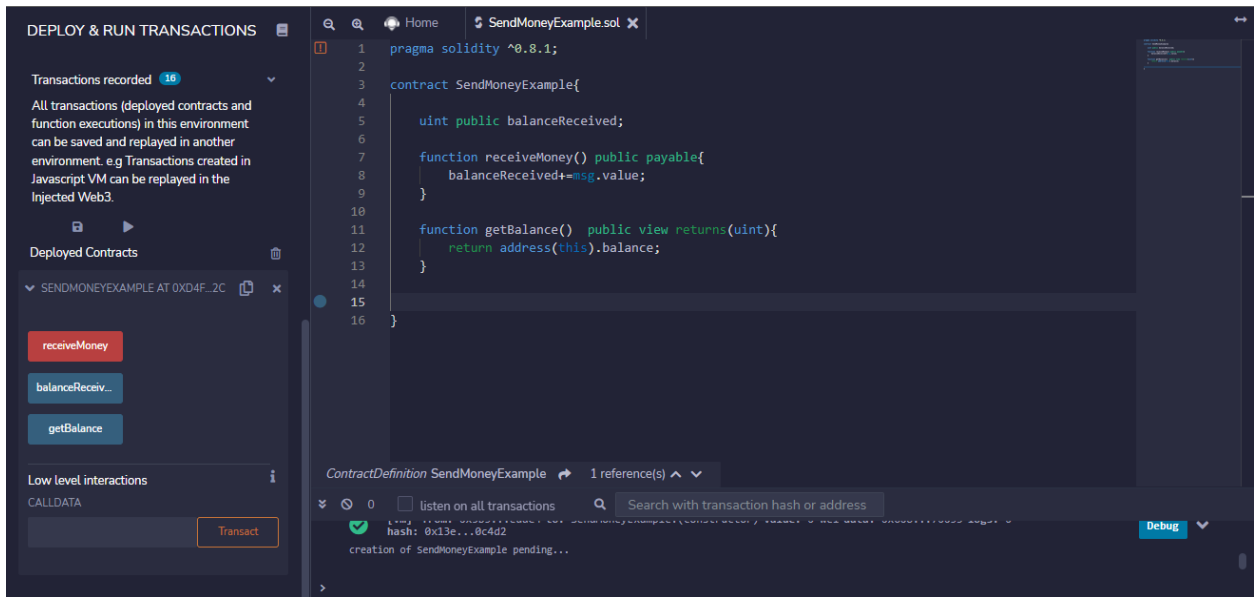
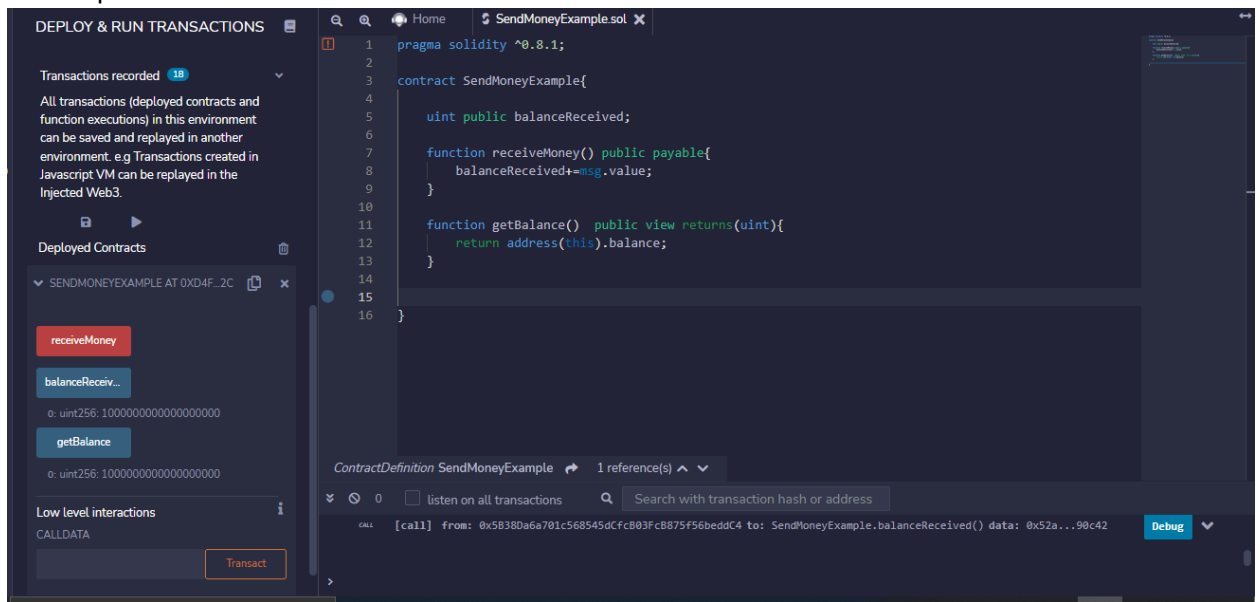


Technical Documentation for Lab 1 Reproduce (Deposit / Withdraw Ether)

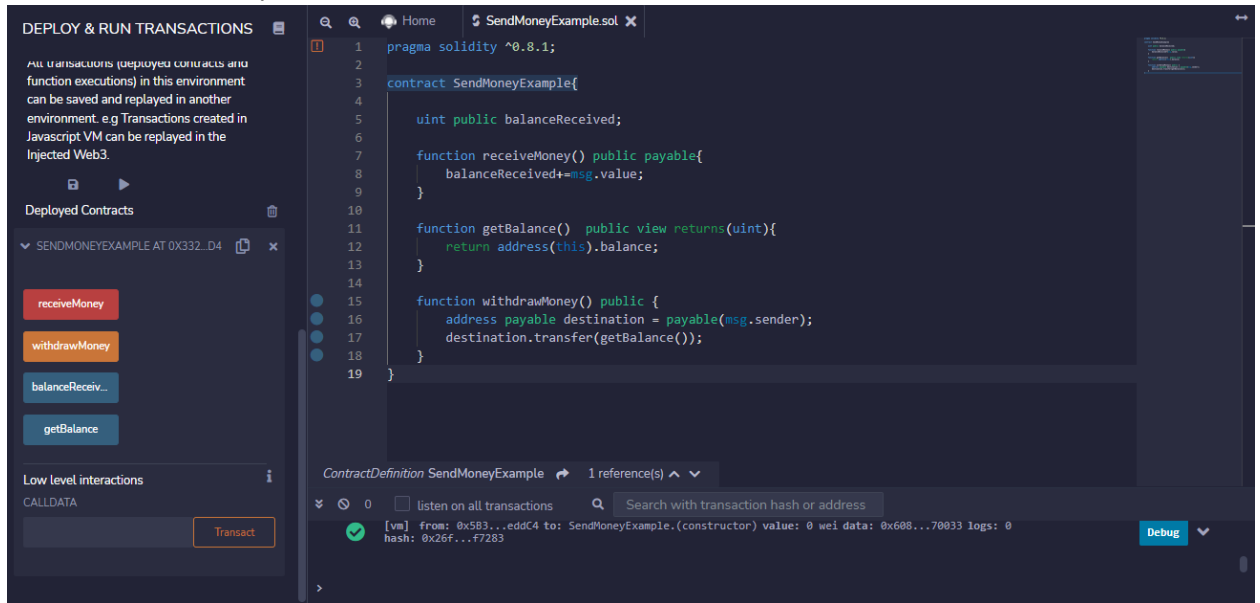
1. Create Initial Smart Contract with basic functionalities :



2. Test Deposit Ethereum :



3. Add Withdraw Money Function :

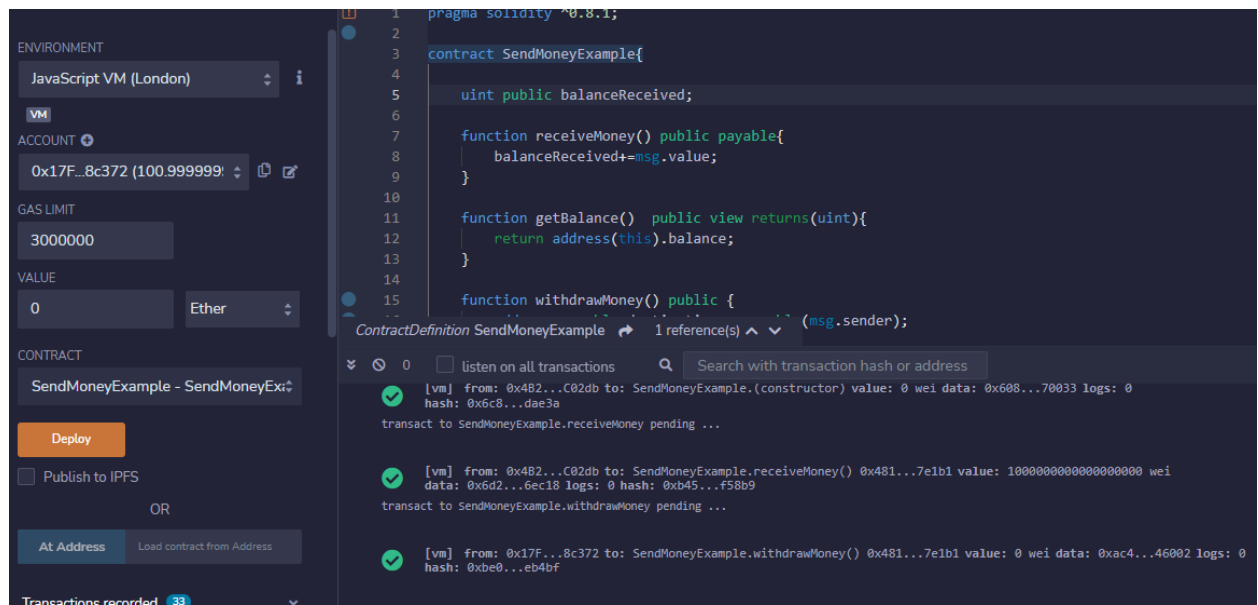


The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel displays the 'SENDMONEYEXAMPLE AT 0X332...D4' contract with buttons for 'receiveMoney', 'withdrawMoney', 'balanceReceiv...', and 'getBalance'. The main editor shows the Solidity code for 'SendMoneyExample.sol' with the following functions:

```
1 pragma solidity ^0.8.1;
2
3 contract SendMoneyExample{
4
5     uint public balanceReceived;
6
7     function receiveMoney() public payable{
8         balanceReceived+=msg.value;
9     }
10
11     function getBalance() public view returns(uint){
12         return address(this).balance;
13     }
14
15     function withdrawMoney() public {
16         address payable destination = payable(msg.sender);
17         destination.transfer(getBalance());
18     }
19 }
```

The bottom panel shows the 'ContractDefinition SendMoneyExample' with a search bar and a list of transactions. The first transaction is a successful constructor call: '[vm] from: 0x5B3...eddC4 to: SendMoneyExample.(constructor) value: 0 wei data: 0x608...70033 logs: 0 hash: 0x26f...f7283'.

4. Balance Increased after withdraw to destination



The screenshot shows the Remix IDE interface. On the left, the 'ENVIRONMENT' panel shows 'JavaScript VM (London)' and 'ACCOUNT' '0x17F...8c372 (100.999999)'. The 'CONTRACT' panel shows 'SendMoneyExample - SendMoneyExi' with a 'Deploy' button. The main editor shows the Solidity code for 'SendMoneyExample.sol' with the following functions:

```
1 pragma solidity ^0.8.1;
2
3 contract SendMoneyExample{
4
5     uint public balanceReceived;
6
7     function receiveMoney() public payable{
8         balanceReceived+=msg.value;
9     }
10
11     function getBalance() public view returns(uint){
12         return address(this).balance;
13     }
14
15     function withdrawMoney() public {
16         address payable destination = payable(msg.sender);
17         destination.transfer(getBalance());
18     }
19 }
```

The bottom panel shows the 'ContractDefinition SendMoneyExample' with a search bar and a list of transactions. The first transaction is a successful constructor call: '[vm] from: 0x482...C02db to: SendMoneyExample.(constructor) value: 0 wei data: 0x608...70033 logs: 0 hash: 0x6c8...dae3a'. The second transaction is a successful call to 'receiveMoney': '[vm] from: 0x482...C02db to: SendMoneyExample.receiveMoney() 0x481...7e1b1 value: 1000000000000000000 wei data: 0x6d2...6ec18 logs: 0 hash: 0xb45...f58b9'. The third transaction is a successful call to 'withdrawMoney': '[vm] from: 0x17F...8c372 to: SendMoneyExample.withdrawMoney() 0x481...7e1b1 value: 0 wei data: 0xac4...46002 logs: 0 hash: 0xbe0...eb4bf'.