



**Pimpri Chinchwad Education Trust's Pimpri  
Chinchwad College of Engineering**



**Laboratory Practice III**

**Department: Computer Engineering**

**Academic Year: 2022 -2023**

**Semester: I**

**Year: B.E**

**Div: B**

**Name: Umesh Chaudhari**

**Roll No: BECOB279**

**Title:** Write a smart contract on a test network, for Bank account of a customer for following operations:

- ☐ Deposit money
- ☐ Withdraw Money
- ☐ Show balance

**Code Output:**

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is visible, showing the environment set to 'Remix VM (London)', the account '0x5B3...eddC4 (99.9999999%)', and the gas limit set to '3000000'. The contract 'Bank - contracts/bank.sol' is selected for deployment. The main editor shows the Solidity code for the 'Bank' contract, which includes a constructor, a modifier 'onlyOwner', and functions 'deposit' and 'withdraw'. The bottom panel shows the transaction log with a successful deployment: '[vm] from: 0x5B3...eddC4 to: Bank.(constructor) value: 0 wei data: 0x608...70033 logs: 0 hash: 0x0f6...696d3'.

```
1 //Keyur Kolambe
2 //BECOB219
3 //SPDX-License-Identifier: GPL-3.0
4 pragma solidity >=0.7.0 <0.9.0;
5
6
7 contract Bank{
8     address public owner;
9     mapping(address=>uint256) private userbalance;
10
11     constructor() public {
12         owner=msg.sender;
13     }
14     modifier onlyOwner(){
15         require (msg.sender==owner,'You are not the owner of this contract');
16         _;
17     }
18
19     function deposit() public payable returns(bool){
20         require(msg.value>10 wei, 'Please deposit at least 10 wei');
21         return true;
22     }
23
24     function withdraw(uint256 _amount) public payable returns(bool){
25         require(_amount <= userbalance[msg.sender], 'You dont have sufficient funds');
26         userbalance[msg.sender]-=_amount;
27         payable(msg.sender).transfer(_amount);
28     }
29 }
```

[vm] from: 0x5B3...eddC4 to: Bank.(constructor) value: 0 wei data: 0x608...70033 logs: 0 hash: 0x0f6...696d3

DEPLOY & RUN TRANSACTIONS

Transactions recorded 1

Deployed Contracts

BANK AT 0XD91...39138 (MEMORY)

Balance: 0 ETH

deposit

withdraw uint256 \_amount

withdrawB... uint256 \_amount

getbalance

getBankBa...

owner

Low level interactions

CALLDATA

Transact

bank.sol

```
24 function withdraw(uint256 _amount) public payable returns(bool){
25     require(_amount <= userbalance[msg.sender], 'You dont have sufficient funds');
26     userbalance[msg.sender]-=_amount;
27     payable(msg.sender).transfer(_amount);
28     return true;
29 }
30
31 function getbalance() public view returns(uint256){
32     return userbalance[msg.sender];
33 }
34
35 function getBankBalance() public view onlyOwner returns(uint256){
36     return address(this).balance;
37 }
38
39 function withdrawBankBalance(uint256 _amount) public payable onlyOwner returns(bool){
40     payable(owner).transfer(_amount);
41     return true;
42 }
43
44 }
```

0

☐ listen on all transactions

Search with transaction hash or address

[vm] from: 0x583...eddC4 to: Bank.(constructor) value: 0 wei data: 0x608...70033 logs: 0

hash: 0x0f6...696d3

Debug