

BC220 BT ASSIGNMENT: - 3

Program –

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.2 <0.9.0;
// Write a smart contract on a test network, for Bank account of a customer for
// following operations: Deposit money | Withdraw Money | Show balance
contract demo{
mapping(address => uint) public user_account;
mapping(address => bool) public user_exist;

function create_account() public payable returns(string memory){
require(user_exist[msg.sender] == false, "Account Already created!");
user_account[msg.sender] = msg.value;
user_exist[msg.sender] = true;
return "Account created";
}

function deposit(uint amount) public payable returns(string memory){
require(user_exist[msg.sender] == true, "Account not created!");
require(amount > 0, "Amount should be greater than 0");
user_account[msg.sender] += amount;
return "Amount deposited sucessfully";
}

function withdraw(uint amount) public payable returns(string memory){
require(user_exist[msg.sender] == true, "Account not created!");
require(amount > 0, "Amount should be greater than 0");
require(user_account[msg.sender] >= amount, "Amount is greater than money deposited");
user_account[msg.sender] -= amount;
return "Amount withdrawn sucessfully";
}

function account_balance() public view returns(uint){
return user_account[msg.sender];
}
function account_exists() public view returns(bool){
return user_exist[msg.sender];
}
}
```

Output –

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is active, showing the 'SmartContract.sol' file. The 'Deploy' button is highlighted. Below it, the 'Deployed Contracts' section shows a contract named 'DEMO AT 0xD4F...2CBE' with a balance of 0 ETH. The 'create_account' button is visible. The main editor shows the Solidity code for 'SmartContract.sol', which includes functions for creating an account, depositing, withdrawing, and checking account balance and existence.

```
1 // SPDX-License-Identifier: GPL-3.0
2
3
4 pragma solidity >=0.8.2 <0.9.0;
5 // Write a smart contract on a test network, for Bank account of a customer for
6 // following operations: Deposit money | Withdraw Money | Show balance
7 contract demo{
8     mapping(address => uint) public user_account;
9     mapping(address => bool) public user_exist;
10
11     function create_account() public payable returns(string memory){ @infinite gas
12         require(user_exist[msg.sender] == false, "Account Already created!");
13         user_account[msg.sender] = msg.value;
14         user_exist[msg.sender] = true;
15         return "Account created";
16     }
17
18     function deposit(uint amount) public payable returns(string memory){ @infinite gas
19         require(user_exist[msg.sender] == true, "Account not created!");
20         require(amount > 0, "Amount should be greater than 0");
21         user_account[msg.sender] += amount;
22         return "Amount deposited successfully";
23     }
24
25     function withdraw(uint amount) public payable returns(string memory){ @infinite gas
26         require(user_exist[msg.sender] == true, "Account not created!");
27         require(amount > 0, "Amount should be greater than 0");
28         require(user_account[msg.sender] >= amount, "Amount is greater than money deposited");
29         user_account[msg.sender] -= amount;
30         return "Amount withdrawn successfully";
31     }
32
33     function account_balance() public view returns(uint){ @2570 gas
34         return user_account[msg.sender];
35     }
36
37     function account_exists() public view returns(bool){ @2591 gas
38         return user_exist[msg.sender];
39     }
40 }
41
```

The screenshot displays the Remix IDE interface after several transactions have been executed. The 'DEPLOY & RUN TRANSACTIONS' sidebar is active, showing the 'SmartContract.sol' file. The 'Deployed Contracts' section shows the contract 'DEMO AT 0xD4F...2CBE' with a balance of 0 ETH. The 'deposit' and 'withdraw' buttons are visible. The main editor shows the Solidity code for 'SmartContract.sol'. The bottom panel shows the debug logs, which include the following transactions:

- [vm] from: 0x5B3...eddC4 to: demo.create_account() 0xD4F...2CBE value: 0 wei data: 0x509...f8633 logs: 0 hash: 0xb7b...ff82e
- [vm] from: 0x5B3...eddC4 to: demo.deposit(uint256) 0xD4F...2CBE value: 0 wei data: 0xb6b...02710 logs: 0 hash: 0xb8c...cfa8f
- [vm] from: 0x5B3...eddC4 to: demo.withdraw(uint256) 0xD4F...2CBE value: 0 wei data: 0x2e1...00fa0 logs: 0 hash: 0x9d...19710
- [call] from: 0x5B38da6a701c568545dcf803fc8075f56beddC4 to: demo.account_balance() data: 0xc08...b9399
- [call] from: 0x5B38da6a701c568545dcf803fc8075f56beddC4 to: demo.account_exists() data: 0x33f...5915b