

**Name** Vedant  
**Roll No.** COBB033  
**Subject:** BT












```
pragma solidity ^0.6;
contract banking
{
    mapping(address=>uint) public user_account;
    mapping(address=>bool) public user_exists;
    function create_account() public payable returns(string memory)
    {
        require(user_exists[msg.sender]==false,'Account already created');
        if(msg.value==0)
        {
            user_account[msg.sender]=0;
            user_exists[msg.sender]=true;
            return "Account created";
        }
        require(user_exists[msg.sender]==false,"Account already created");
        user_account[msg.sender]=msg.value;
        user_exists[msg.sender]=true; return
        "Account created";
    }
    function deposit() public payable returns(string memory)
    {
        require(user_exists[msg.sender]==true,"Account not created");
        require(msg.value>0,"Value for deposit is Zero");
        user_account[msg.sender]=user_account[msg.sender]+msg.value;
        return "Deposited Successfully";
    }
    function withdraw(uint amount) public payable returns(string memory)
    {
        require(user_account[msg.sender]>amount,"Insufficient Balance");
        require(user_exists[msg.sender]==true,"Account not created");
        require(amount>0,"Amount should be more than zero");
        user_account[msg.sender]=user_account[msg.sender]-amount;
        msg.sender.transfer(amount);
        return "Withdrawl Successful";
    }
    function transfer(address payable userAddress, uint amount) public returns(string
    memory)
    {
        require(user_account[msg.sender]>amount,"Insufficient balance in Bank
        account"); require(user_exists[msg.sender]==true,"Account is not created");
        require(user_exists[userAddress]==true,"Transfer account does not exist");
        require(amount>0,"Amount should be more than zero");
        user_account[msg.sender]=user_account[msg.sender]-amount;
```

```

user_account[userAddress]=user_account[userAddress]+amount;
return "Transfer Successful";
}
function send_amt(address payable toAddress, uint256 amount) public payable
returns(string memory)
{
require(user_account[msg.sender]>amount,"Insufficeint balance in Bank
account"); require(user_exists[msg.sender]==true,"Account is not created");
require(amount>0,"Amount should be more than zero");
user_account[msg.sender]=user_account[msg.sender]-amount;
toAddress.transfer(amount);
return "Transfer Success";
}
function user_balance() public view returns(uint)
{
return user_account[msg.sender];
}
function account_exist() public view returns(bool)
{
return user_exists[msg.sender];
}
}


```



Compile SOL code    Deploy contract



DEPLOY & RUN  
TRANSACTIONS

VM

ACCOUNT 


0x17F...8c372 (39.9999999)  

GAS LIMIT


3000000

VALUE

0

Ether 

CONTRACT

banking - banking.sol 



evm version: istanbul


Deploy




☐ Publish to IPFS

At Address

Load contract from Address

Transactions recorded 34  

Deployed Contracts 

 BANKING AT 0XC47...F10B2 (MEI)  



# SOLIDITY COMPILER



COMPILER +

0.6.12+commit.27d51765



☐ Include nightly builds

☐ Auto compile

☐ Hide warnings

Advanced Configurations



Compile aj.sol

Compile and Run script



CONTRACT

Banking (aj.sol)



Publish on Ipfs



Publish on Swarm



Compilation Details



ABI



Bytecode

The screenshot displays the Etherscan interface for a contract named 'BANKING AT 0XC47...F10B2'. The 'Deployed Contracts' section is active, showing the 'create\_account' function. The right pane provides detailed information about the transaction, including the transaction hash, block hash, block number, from address, to address, gas, transaction cost, execution cost, input, decoded input, decoded output, logs, and val.

Field	Value
transaction hash	0x997c7a7018b70b082751175f4427cd8b9bc7c78fa30ea2e8ed96b083afc7f01
block hash	0x508c4eb253719b40f50913995b6e0acdf9b50c20843b74f52520bdc068955
block number	35
from	0x17f6AD8EF9822297579C283049C10bffeA348c372
to	banking.create_account() 0xc47e70c386290e40E56128425e27c50f8cf1002
gas	53562 gas
transaction cost	46575 gas
execution cost	25511 gas
input	0x509...f8633
decoded input	{}
decoded output	{ "0": "string: Account created" }
logs	[ ]
val	0 wei

### Deposit Amount

## Send Amount

## Check Account Exists

The screenshot shows a web interface with a sidebar on the left and a main content area on the right. The sidebar contains several buttons: **account\_exist**, **user\_account**, **user\_balance**, and **user\_exists**. Below these is a section titled "Low level interactions" with a "CALLDATA" input field and a "Transact" button. The main content area displays transaction details for the **account\_exist** function. The details include:   
- **to**: banking.account\_exist() 0xC47e78C386290E48E561284255e27c50f8Cf10b2   
- **execution cost**: 2494 gas (Cost only applies when called by a contract)   
- **input**: 0xcde...6e57b   
- **decoded input**: {}   
- **decoded output**: {"0": "bool: true"}   
- **logs**: []

## Check User balance

The screenshot shows the same web interface as the previous one, but with the **user\_balance** button selected. The sidebar now shows **user\_balance** as the active button, and the "Low level interactions" section is still visible. The main content area displays transaction details for the **user\_balance** function. The details include:   
- **decoded input**: {}   
- **decoded output**: {"0": "uint256: 3999999999999999995"}   
- **logs**: []

# Withdraw Amount

transferaddress userAddress, uint2

withdraw

amount: "50"

CalldataParameterstransact

account\_exist

user\_accountaddress

listen on all transactions

Search with transaction hash or address

execution cost15250 gas

input0x2e1...00032

decoded input{"uint256 amount": "50"}

decoded output{"0": "string: Withdrawal Successful"}

logs[]