PRACTICAL 9

[CS601] – Cryptography and Blockchain

Date – 23/03/2023 | *By* Aishwarya Suryakant Waghmare, PRN – 2001106059

Title/Aim of the practical :

To write smart contracts in the solidity programming regarding the Advanced Solidity : Quantifiers, constructor, override, abstract, inheritance as well as error handling and solving the following exercises as well given below :

- ✓ Create an abstract base contract called Calculator with a read-only public function that returns integers.
- ✓ Create a derived contract called Test which derives the Calculator contract and can calculate 1 + 2 and return the result.

Apparatus/Tools/ Resources used :

- Lecture Notes
- E-Resources
- E-Book
- Laptop
- Remix IDE

Procedure of the practical/ Program Code :

To write a smart contract in the solidity programming regarding the following :

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract C {
   uint256 internal data;
   uint256 public info;

   constructor() {
      info = 10;
   }

   function increment(uint256 a) internal pure returns (uint256) {
      return a + 1;
   }

   function updateData(uint256 a) public {
        data = a;
    }

    function getData() public view returns (uint256) {
        return data;
    }

    function compute(uint256 a, uint256 b) internal pure returns (uint256) {
        return a + b;
    }
}

contract D {
    C private c = new C();

    function readInfo() public view returns (uint256) {
        return c.info();
    }
}

contract E is C {
    uint256 private result;
    C private c;

    constructor() {
        c = new C();
    }

    function getComputedResult() public {
        result = compute(23, 5);
    }

    function getResult() public view returns (uint256) {
        return result;
    }

    function getInfo() public view returns (uint256) {
        return c.info();
    }
}
```

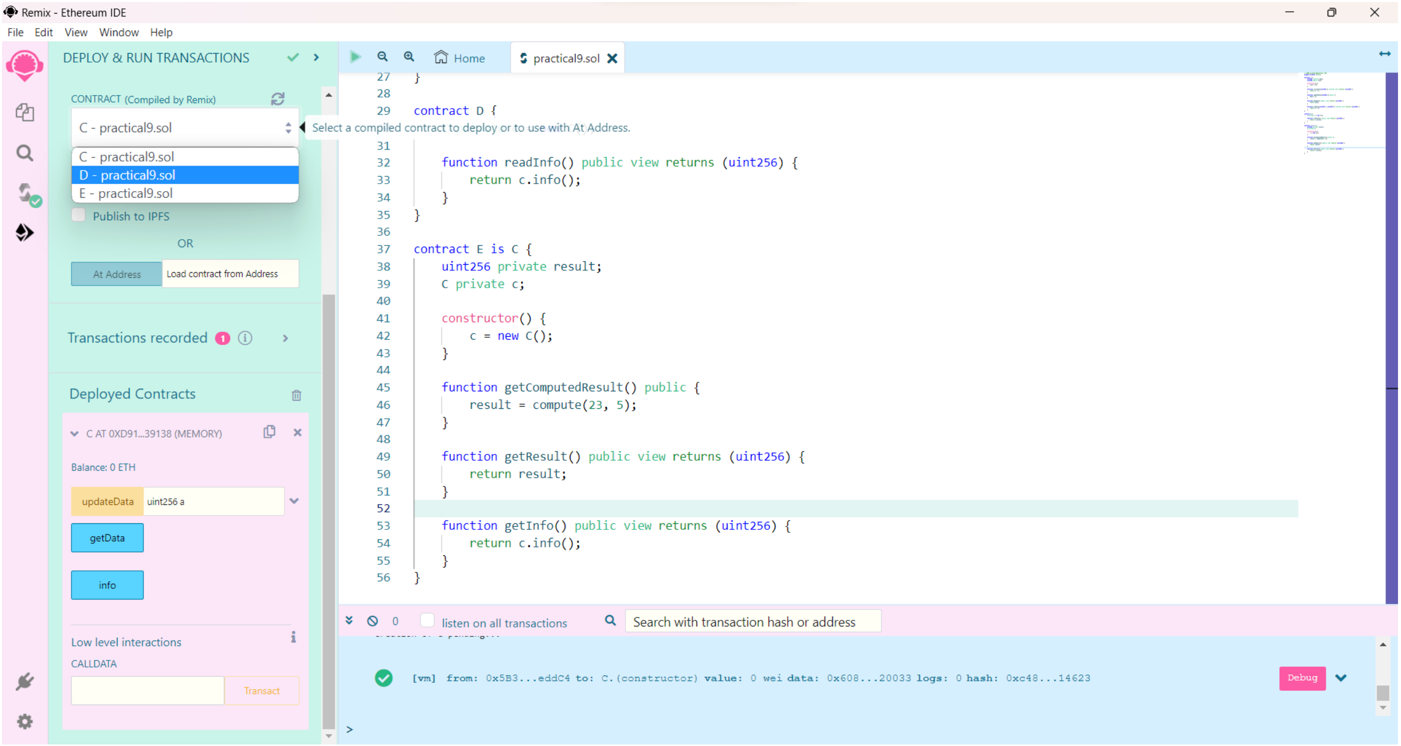
Solidity Error handling code :

```
pragma solidity ^0.8.0;

contract MyContract {
    address public owner;
    uint public balance;

    event ErrorOccurred(string errorMessage);

    constructor() {
        owner = msg.sender;
        balance = 0;
    }

    function deposit() public payable {
        balance += msg.value;
    }

    function withdraw(uint amount) public {
        require(msg.sender == owner, "Only owner can withdraw");
        require(amount <= balance, "Insufficient balance");

        balance -= amount;
        (bool success, ) = msg.sender.call{value: amount}("");
        if (!success) {
            emit ErrorOccurred("Failed to send ether");
            balance += amount;
        }

    }

}
```

Exercises 1 and 2 :

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.5.0 <0.9.0;
abstract contract Calculator {
    function getResult() public view virtual returns (uint256);
}

// SPDX-License-Identifier: MIT
pragma solidity >=0.5.0 <0.9.0;
abstract contract Calculator {
    function getResult() public view virtual returns (uint256);
}

contract Test is Calculator {
    function getResult() public pure override returns (uint256) {
        return 1 + 2;
    }
}
```

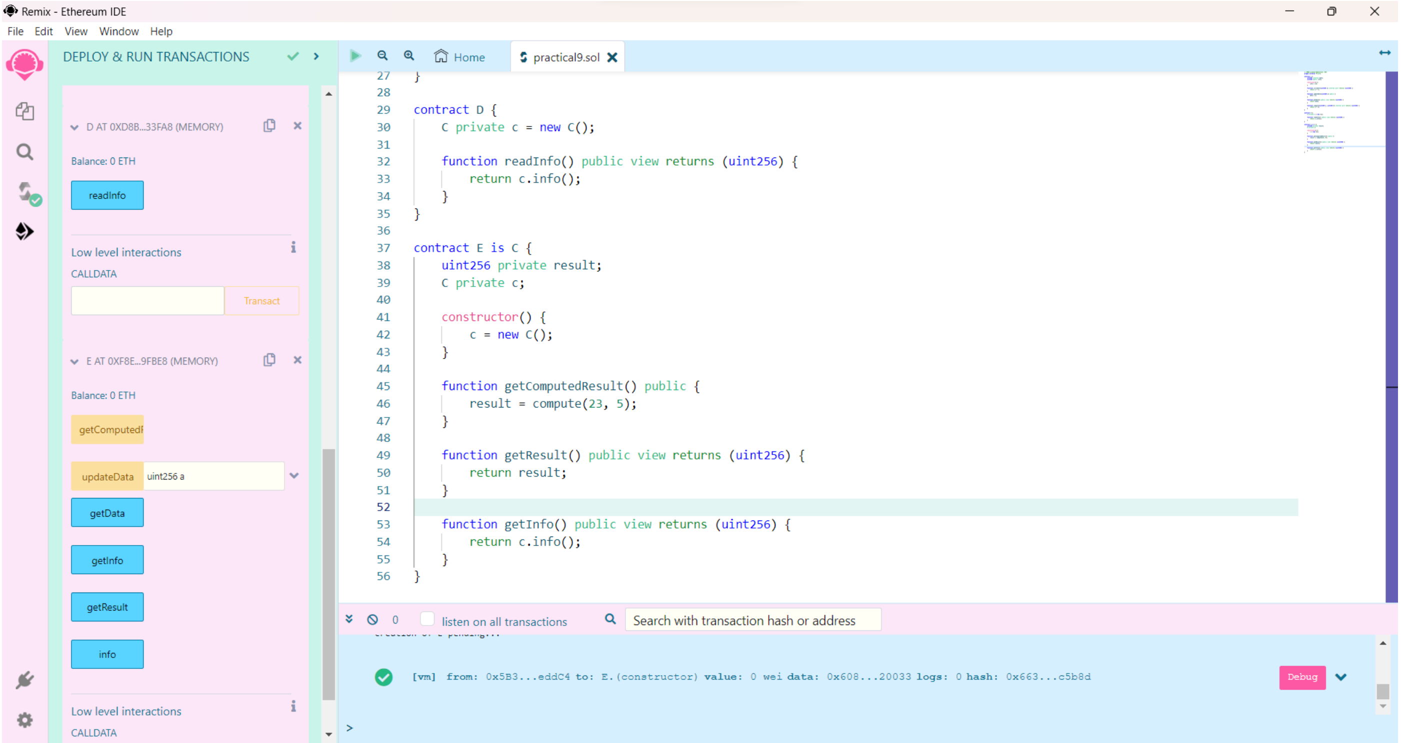
Result/ Output/ Screenshots of the practical :



Select a compiled contract to deploy or to use with At Address.

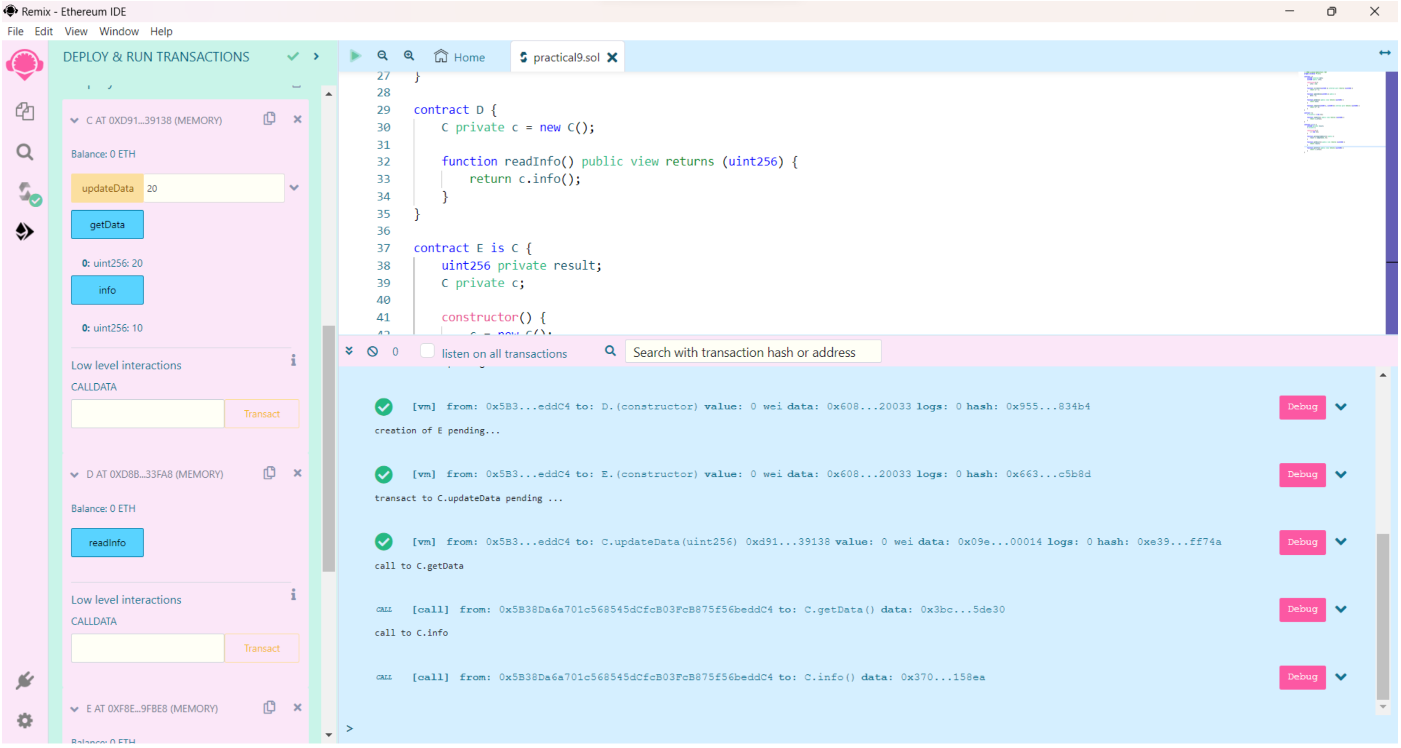
```
}

contract D {

    function readInfo() public view returns (uint256) {
        return c.info();
    }
}

contract E is C {
    uint256 private result;
    C private c;

    constructor() {
        c = new C();
    }

    function getComputedResult() public {
        result = compute(23, 5);
    }

    function getResult() public view returns (uint256) {
        return result;
    }

    function getInfo() public view returns (uint256) {
        return c.info();
    }
}
```

```
[vm] from: 0x5B3...eddC4 to: C.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0xc48...14623
```



```
}

contract D {
    C private c = new C();

    function readInfo() public view returns (uint256) {
        return c.info();
    }
}

contract E is C {
    uint256 private result;
    C private c;

    constructor() {
        c = new C();
    }

    function getComputedResult() public {
        result = compute(23, 5);
    }

    function getResult() public view returns (uint256) {
        return result;
    }

    function getInfo() public view returns (uint256) {
        return c.info();
    }
}
```

```
[vm] from: 0x5B3...eddC4 to: E.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x663...c5b8d
```

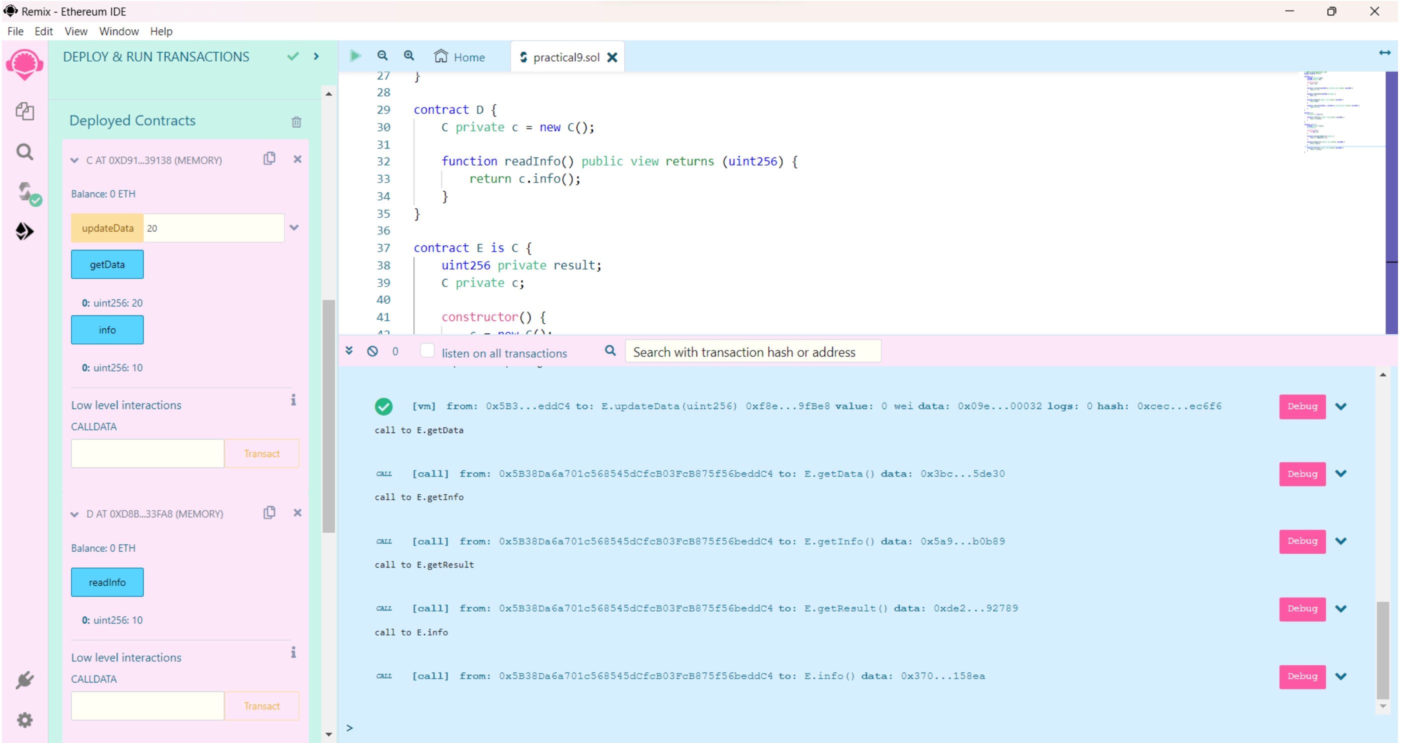

Remix - Ethereum IDE
File Edit View Window Help
DEPLOY & RUN TRANSACTIONS
Home
practical9.sol
C AT 0XD91...39138 (MEMORY)
Balance: 0 ETH
updateData
20
getData
info
Low level interactions
CALLDATA
Transact
D AT 0XD8B...33FA8 (MEMORY)
Balance: 0 ETH
readInfo
Low level interactions
CALLDATA
Transact
E AT 0XF8E...9FBE8 (MEMORY)
0
listen on all transactions
Search with transaction hash or address
Debug

0: uint256: 20

0: uint256: 10

```
}

contract D {
    C private c = new C();

    function readInfo() public view returns (uint256) {
        return c.info();
    }
}

contract E is C {
    uint256 private result;
    C private c;

    constructor() {
```

```
[vm] from: 0x5B3...eddC4 to: D.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x955...834b4
creation of E pending...

[vm] from: 0x5B3...eddC4 to: E.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x663...c5b8d
transact to C.updateData pending ...

[vm] from: 0x5B3...eddC4 to: C.updateData(uint256) 0xd91...39138 value: 0 wei data: 0x09e...00014 logs: 0 hash: 0xe39...ff74a
call to C.getData

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: C.getData() data: 0x3bc...5de30
call to C.info

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: C.info() data: 0x370...158ea
```



updateData 20

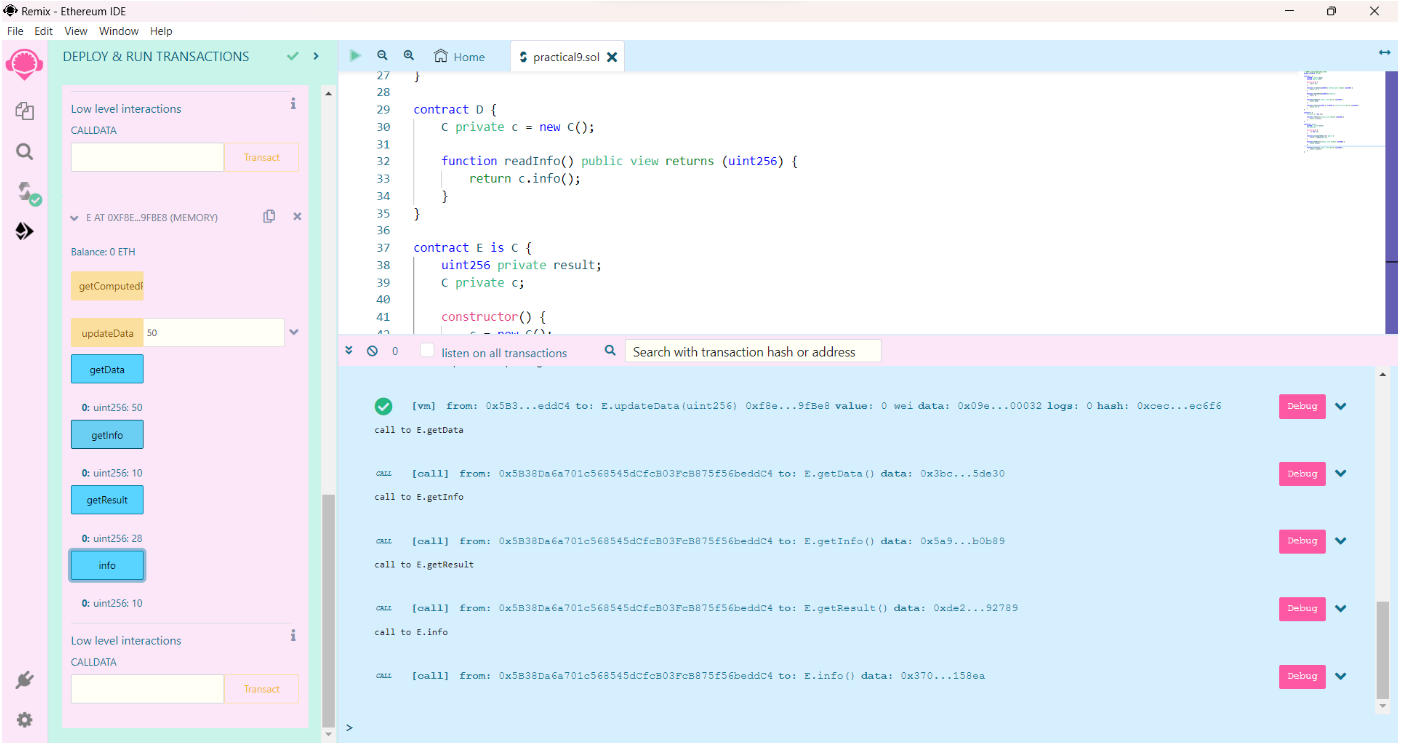
0: uint256: 20

0: uint256: 10

0: uint256: 10

```
}

contract D {
    C private c = new C();

    function readInfo() public view returns (uint256) {
        return c.info();
    }
}

contract E is C {
    uint256 private result;
    C private c;

    constructor() {
```

```
[vm] from: 0x5B3...eddC4 to: E.updateData(uint256) 0xf8e...9fBe8 value: 0 wei data: 0x09e...00032 logs: 0 hash: 0xcec...ec6f6
call to E.getData

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.getData() data: 0x3bc...5de30
call to E.getInfo

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.getInfo() data: 0x5a9...b0b89
call to E.getResult

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.getResult() data: 0xde2...92789
call to E.info

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.info() data: 0x370...158ea

>
```



updateData 50

0: uint256: 50

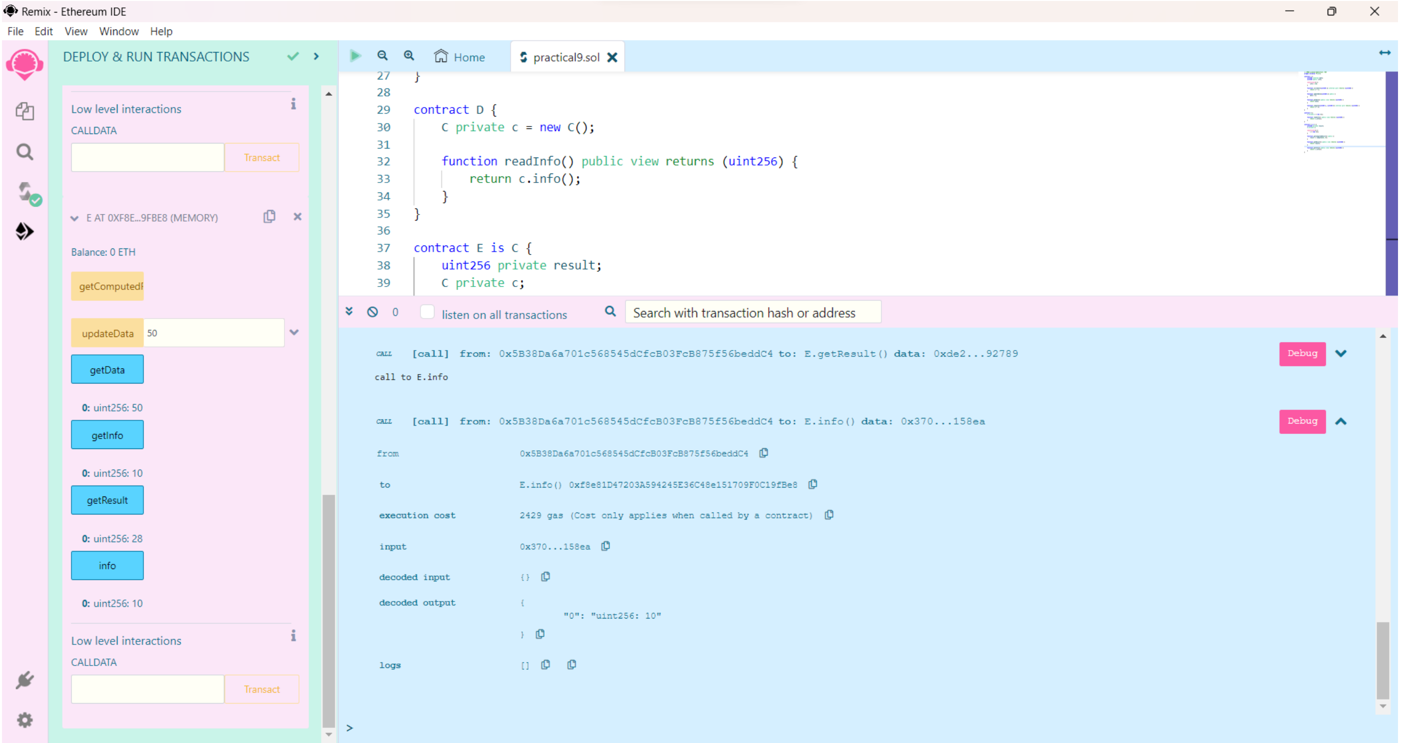
0: uint256: 10

0: uint256: 28

0: uint256: 10

```
}

contract D {
    C private c = new C();

    function readInfo() public view returns (uint256) {
        return c.info();
    }
}

contract E is C {
    uint256 private result;
    C private c;

    constructor() {
```

```
[vm] from: 0x5B3...eddC4 to: E.updateData(uint256) 0xf8e...9fBe8 value: 0 wei data: 0x09e...00032 logs: 0 hash: 0xcec...ec6f6
call to E.getData

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.getData() data: 0x3bc...5de30
call to E.getInfo

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.getInfo() data: 0x5a9...b0b89
call to E.getResult

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.getResult() data: 0xde2...92789
call to E.info

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.info() data: 0x370...158ea

>
```



Low level interactions
CALLDATA
Transact

E AT 0XF8E...9FBE8 (MEMORY)

Balance: 0 ETH

getComputedI

updateData 50

getData

0: uint256: 50

getInfo

0: uint256: 10

getResult

0: uint256: 28

info

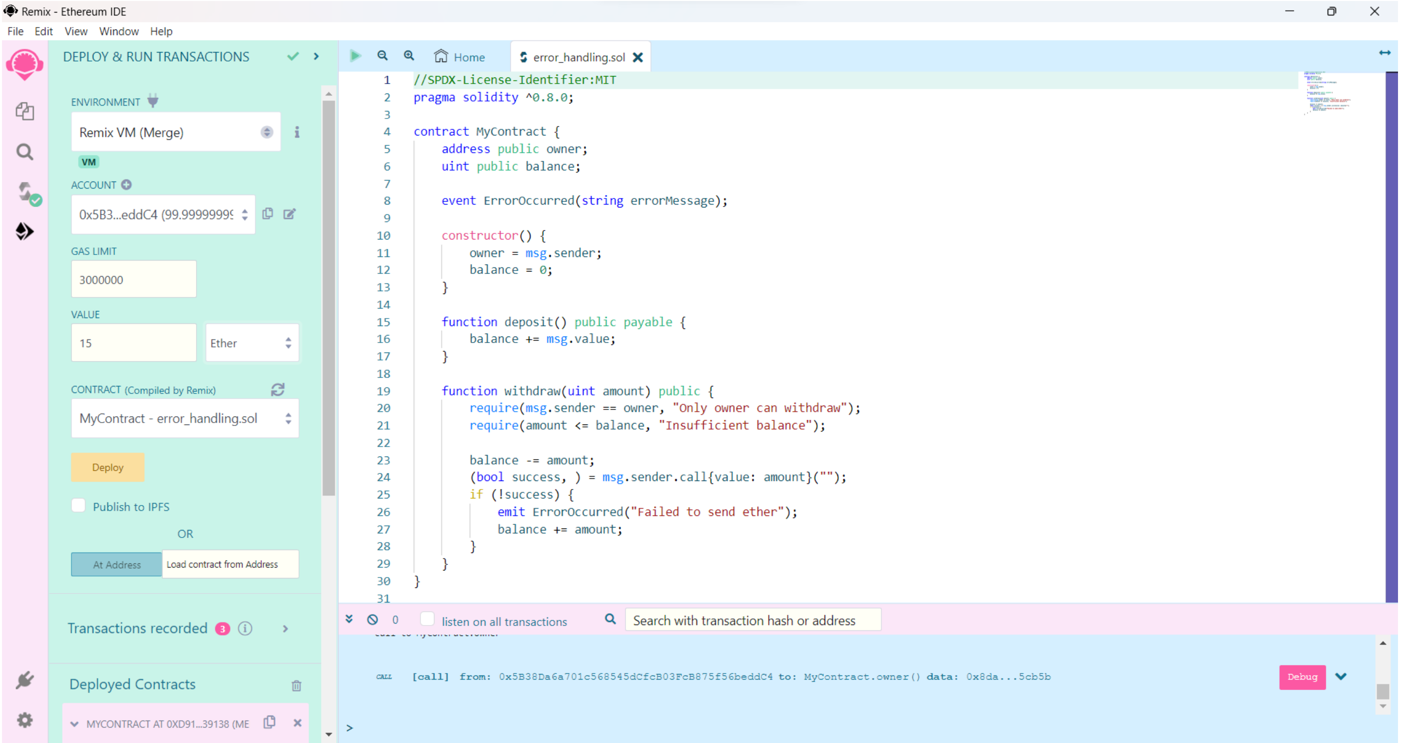
0: uint256: 10

Low level interactions
CALLDATA
Transact

```
}

contract D {
    C private c = new C();

    function readInfo() public view returns (uint256) {
        return c.info();
    }
}

contract E is C {
    uint256 private result;
    C private c;
```

0 ☐ listen on all transactions 　Search with transaction hash or address

```
CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.getResult() data: 0xde2...92789
call to E.info

CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: E.info() data: 0x370...158ea
from              0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                E.info() 0xf8e81D47203A594245E36C48e151709F0C19fBe8
execution cost    2429 gas (Cost only applies when called by a contract)
input             0x370...158ea
decoded input     {}
decoded output    {
                        "0": "uint256: 10"
                  }
logs              []
>
```



ENVIRONMENT
Remix VM (Merge)
VM

ACCOUNT
0x5B3...eddC4 (99.9999999

GAS LIMIT
3000000

VALUE
15 Ether

CONTRACT (Compiled by Remix)
MyContract - error_handling.sol

Deploy

☐ Publish to IPFS

OR

At Address　Load contract from Address

Transactions recorded 3

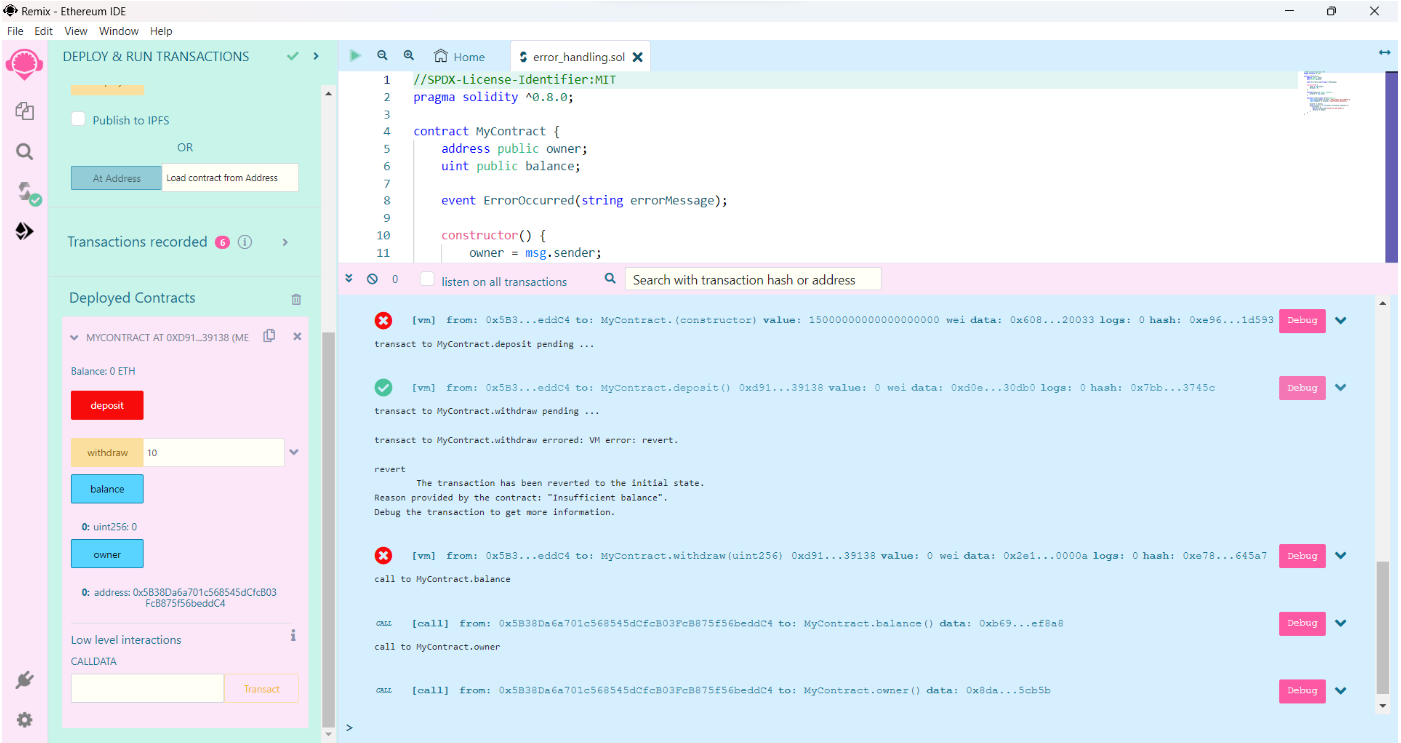
Deployed Contracts

MYCONTRACT AT 0XD91...39138 (ME

```
//SPDX-License-Identifier:MIT
pragma solidity ^0.8.0;

contract MyContract {
    address public owner;
    uint public balance;

    event ErrorOccurred(string errorMessage);

    constructor() {
        owner = msg.sender;
        balance = 0;
    }

    function deposit() public payable {
        balance += msg.value;
    }

    function withdraw(uint amount) public {
        require(msg.sender == owner, "Only owner can withdraw");
        require(amount <= balance, "Insufficient balance");

        balance -= amount;
        (bool success, ) = msg.sender.call{value: amount}("");
        if (!success) {
            emit ErrorOccurred("Failed to send ether");
            balance += amount;
        }
    }
}

```

0 ☐ listen on all transactions 　Search with transaction hash or address

```
CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: MyContract.owner() data: 0x8da...5cb5b
>
```



Balance: 0 ETH

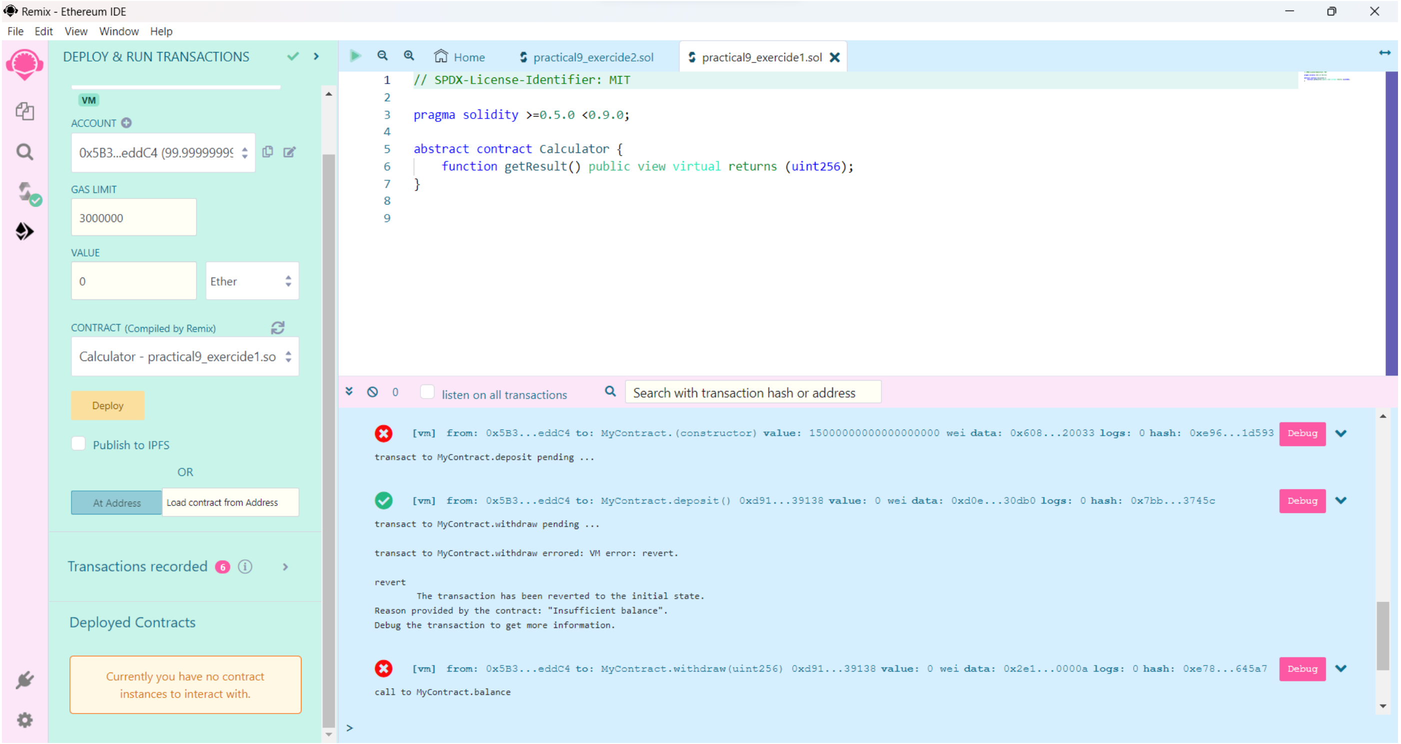
withdraw: 10

balance → 0: uint256: 0

owner → 0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

```solidity
//SPDX-License-Identifier:MIT
pragma solidity ^0.8.0;

contract MyContract {
    address public owner;
    uint public balance;

    event ErrorOccurred(string errorMessage);

    constructor() {
        owner = msg.sender;
```

```
[vm] from: 0x5B3...eddC4 to: MyContract.(constructor) value: 15000000000000000000 wei data: 0x608...20033 logs: 0 hash: 0xe96...1d593
transact to MyContract.deposit pending ...

[vm] from: 0x5B3...eddC4 to: MyContract.deposit() 0xd91...39138 value: 0 wei data: 0xd0e...30db0 logs: 0 hash: 0x7bb...3745c
transact to MyContract.withdraw pending ...

transact to MyContract.withdraw errored: VM error: revert.

revert
	The transaction has been reverted to the initial state.
Reason provided by the contract: "Insufficient balance".
Debug the transaction to get more information.

[vm] from: 0x5B3...eddC4 to: MyContract.withdraw(uint256) 0xd91...39138 value: 0 wei data: 0x2e1...0000a logs: 0 hash: 0xe78...645a7
call to MyContract.balance

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: MyContract.balance() data: 0xb69...ef8a8
call to MyContract.owner

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: MyContract.owner() data: 0x8da...5cb5b
>
```



ACCOUNT: 0x5B3...eddC4 (99.99999999

GAS LIMIT: 3000000

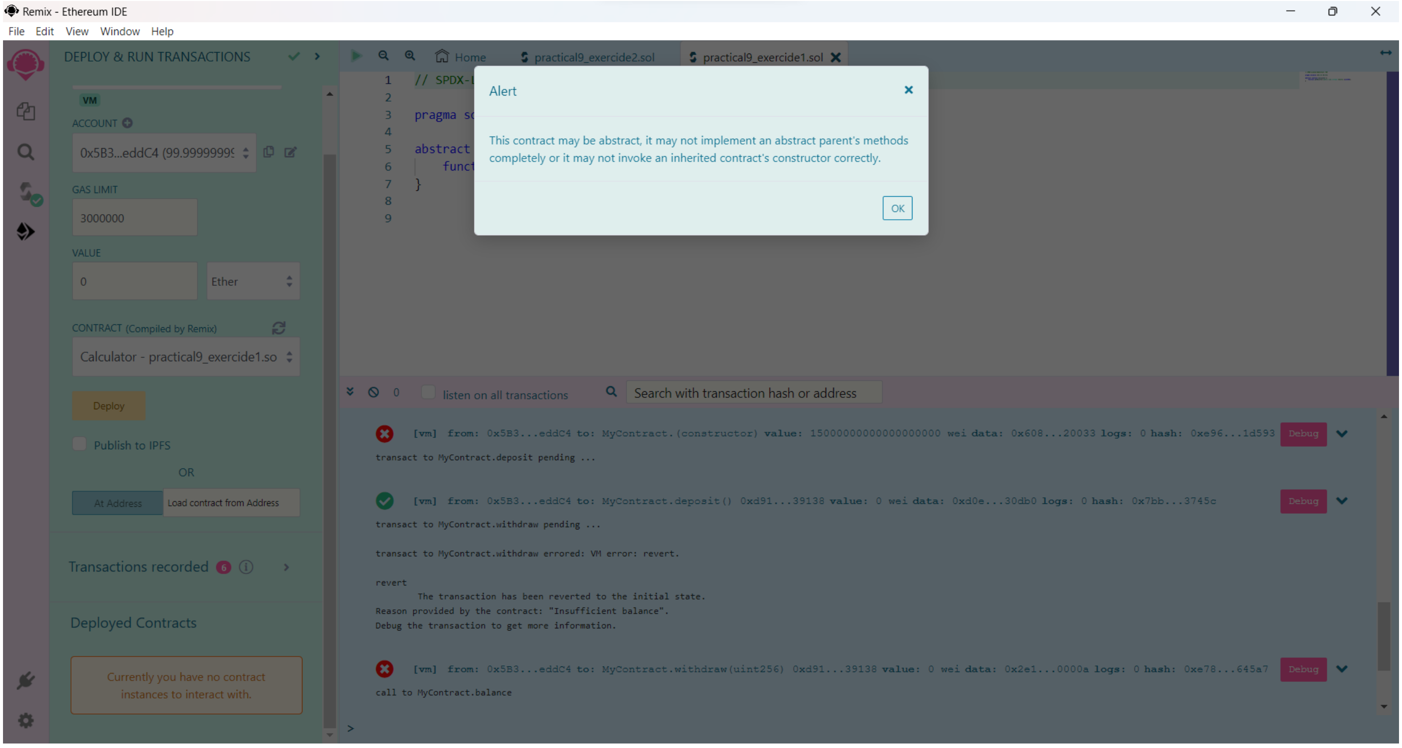
VALUE: 0 Ether

CONTRACT (Compiled by Remix): Calculator - practical9_exercide1.so

Currently you have no contract instances to interact with.

```solidity
// SPDX-License-Identifier: MIT

pragma solidity >=0.5.0 <0.9.0;

abstract contract Calculator {
    function getResult() public view virtual returns (uint256);
}
```

```
[vm] from: 0x5B3...eddC4 to: MyContract.(constructor) value: 15000000000000000000 wei data: 0x608...20033 logs: 0 hash: 0xe96...1d593
transact to MyContract.deposit pending ...

[vm] from: 0x5B3...eddC4 to: MyContract.deposit() 0xd91...39138 value: 0 wei data: 0xd0e...30db0 logs: 0 hash: 0x7bb...3745c
transact to MyContract.withdraw pending ...

transact to MyContract.withdraw errored: VM error: revert.

revert
	The transaction has been reverted to the initial state.
Reason provided by the contract: "Insufficient balance".
Debug the transaction to get more information.

[vm] from: 0x5B3...eddC4 to: MyContract.withdraw(uint256) 0xd91...39138 value: 0 wei data: 0x2e1...0000a logs: 0 hash: 0xe78...645a7
call to MyContract.balance
>
```


Remix - Ethereum IDE
File Edit View Window Help
DEPLOY & RUN TRANSACTIONS
VM
Home
practical9_exercide2.sol
practical9_exercide1.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 6
Deployed Contracts
Currently you have no contract instances to interact with.
listen on all transactions
Search with transaction hash or address
Debug
OK

ACCOUNT 0x5B3...eddC4 (99.99999999

GAS LIMIT 3000000

VALUE 0 Ether

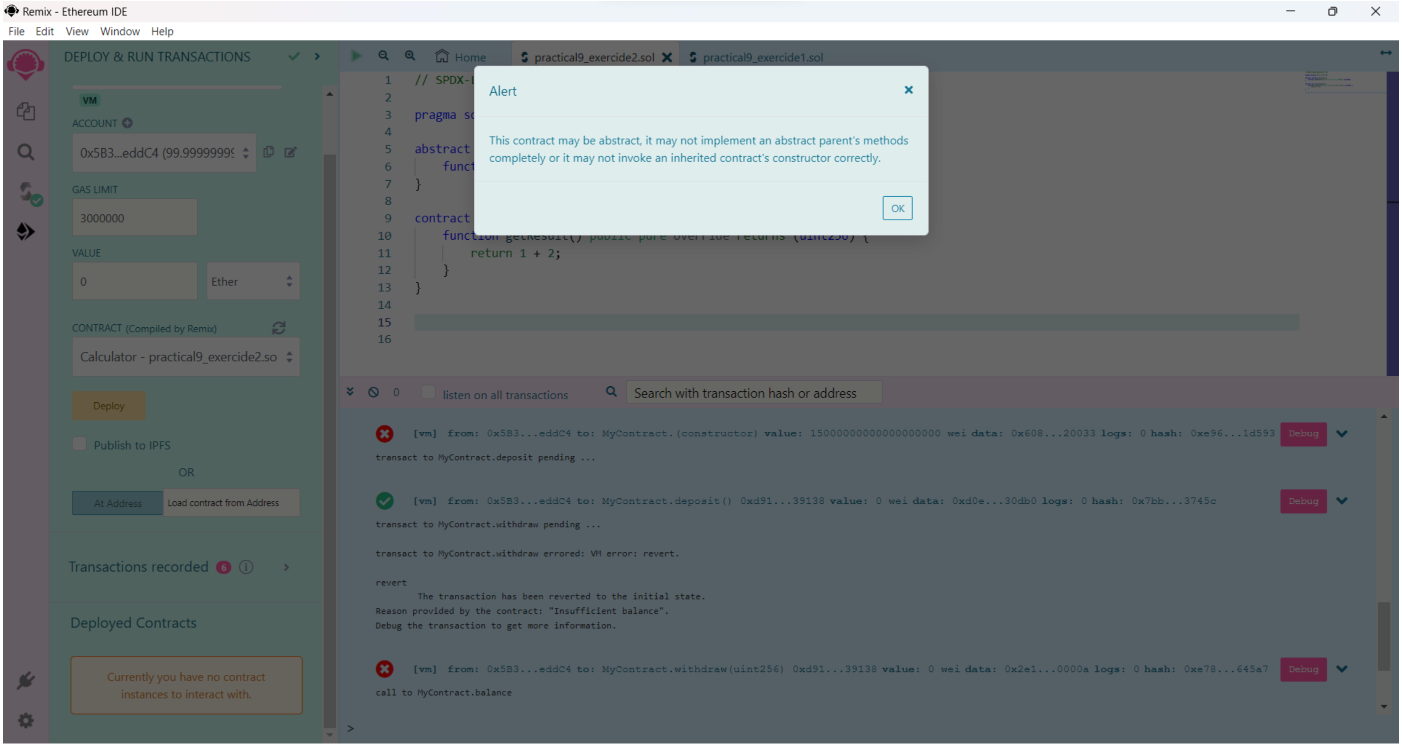
CONTRACT (Compiled by Remix) Calculator - practical9_exercide1.so

```
// SPDX-L

pragma sc

abstract
    funct
}
```

Alert

This contract may be abstract, it may not implement an abstract parent's methods completely or it may not invoke an inherited contract's constructor correctly.

```
[vm] from: 0x5B3...eddC4 to: MyContract.(constructor) value: 15000000000000000000 wei data: 0x608...20033 logs: 0 hash: 0xe96...1d593
transact to MyContract.deposit pending ...

[vm] from: 0x5B3...eddC4 to: MyContract.deposit() 0xd91...39138 value: 0 wei data: 0xd0e...30db0 logs: 0 hash: 0x7bb...3745c
transact to MyContract.withdraw pending ...

transact to MyContract.withdraw errored: VM error: revert.

revert
	The transaction has been reverted to the initial state.
Reason provided by the contract: "Insufficient balance".
Debug the transaction to get more information.

[vm] from: 0x5B3...eddC4 to: MyContract.withdraw(uint256) 0xd91...39138 value: 0 wei data: 0x2e1...0000a logs: 0 hash: 0xe78...645a7
call to MyContract.balance
>
```


Remix - Ethereum IDE
File Edit View Window Help
DEPLOY & RUN TRANSACTIONS
VM
Home
practical9_exercide2.sol
practical9_exercide1.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 6
Deployed Contracts
Currently you have no contract instances to interact with.
listen on all transactions
Search with transaction hash or address
Debug
OK

ACCOUNT 0x5B3...eddC4 (99.99999999

GAS LIMIT 3000000

VALUE 0 Ether

CONTRACT (Compiled by Remix) Calculator - practical9_exercide2.so

```
// SPDX-L

pragma sc

abstract
    funct
}

contract
    function getResult() public pure override returns (uint256) {
        return 1 + 2;
    }
}
```

Alert

This contract may be abstract, it may not implement an abstract parent's methods completely or it may not invoke an inherited contract's constructor correctly.

```
[vm] from: 0x5B3...eddC4 to: MyContract.(constructor) value: 15000000000000000000 wei data: 0x608...20033 logs: 0 hash: 0xe96...1d593
transact to MyContract.deposit pending ...

[vm] from: 0x5B3...eddC4 to: MyContract.deposit() 0xd91...39138 value: 0 wei data: 0xd0e...30db0 logs: 0 hash: 0x7bb...3745c
transact to MyContract.withdraw pending ...

transact to MyContract.withdraw errored: VM error: revert.

revert
	The transaction has been reverted to the initial state.
Reason provided by the contract: "Insufficient balance".
Debug the transaction to get more information.

[vm] from: 0x5B3...eddC4 to: MyContract.withdraw(uint256) 0xd91...39138 value: 0 wei data: 0x2e1...0000a logs: 0 hash: 0xe78...645a7
call to MyContract.balance
>
```

Parameters achieved/ Conclusion :

Therefore, understood and wrote smart contracts in the solidity programming regarding the Advanced Solidity : Quantifiers, constructor, override, abstract, inheritance.