**3C Part 1: Contracts & Inheritance**

**📄 Create a file:**

CopyEdit

ContractsInheritance.sol

**✅ Paste this code:**

solidity

CopyEdit

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.17;

contract C {

uint private data;

uint public info;

constructor() {

info = 10;

}

function increment(uint a) private pure returns(uint) {

return a + 1;

}

function updateData(uint a) public {

data = a;

}

function getData() public view returns(uint) {

return data;

}

function compute(uint a, uint b) internal pure returns (uint) {

return a + b;

}

}

contract D {

function readData() public returns(uint) {

C c = new C();

c.updateData(7);

return c.getData();

}

}

contract E is C {

uint private result;

C private c;

constructor() {

c = new C();

}

function getComputedResult() public {

result = compute(3, 6);

}

function getResult() public view returns(uint) {

return result;

}

}

**🧪 Deploy and Test:**

✅ Deploy **D**:

* Call readData()
  + Should return 7.

✅ Deploy **E**:

* Call getComputedResult()
* Then getResult()—should return 9.

✅ Deploy **C** if you want to test updateData() and getData() independently.

🎉 Contracts and inheritance done!

**🎯 3C Part 2: Constructors**

**📄 Create:**

CopyEdit

Constructors.sol

**✅ Paste this:**

solidity

CopyEdit

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.17;

contract Constructors {

string str;

uint amount;

constructor() {

str = "Shlok is learning Solidity";

amount = 10;

}

function constData() public view returns(string memory, uint) {

return (str, amount);

}

}

**🧪 Deploy and Test:**

✅ Deploy Constructors  
✅ Call constData()

* Should return:
  + "Shlok is learning Solidity"
  + 10

**🎯 3C Part 3: Abstract Contracts**

**📄 Create:**

CopyEdit

AbstractContract.sol

**✅ Paste this:**

solidity

CopyEdit

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.17;

abstract contract Main {

function add(uint a, uint b) public virtual pure returns (uint);

}

contract Adder is Main {

function add(uint a, uint b) public override pure returns (uint) {

return a + b;

}

}

**🧪 Deploy and Test:**

✅ Deploy Adder  
✅ Call add(2,3)—should return 5.

✅ Note:

* You **cannot deploy Main** (it’s abstract).

**🎯 3C Part 4: Interfaces**

**📄 Create:**

CopyEdit

InterfaceDemo.sol

**✅ Paste this:**

solidity

CopyEdit

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.17;

interface IAdder {

function add(uint a, uint b) external pure returns(uint);

}

contract AdderContract is IAdder {

function add(uint a, uint b) external pure returns(uint) {

return a + b;

}

}

**🧪 Deploy and Test:**

✅ Deploy AdderContract  
✅ Call add(7,8)—should return 15.