

Lab Session: Smart Contract Interaction

BSc Blockchain, Crypto Economy & NFTs

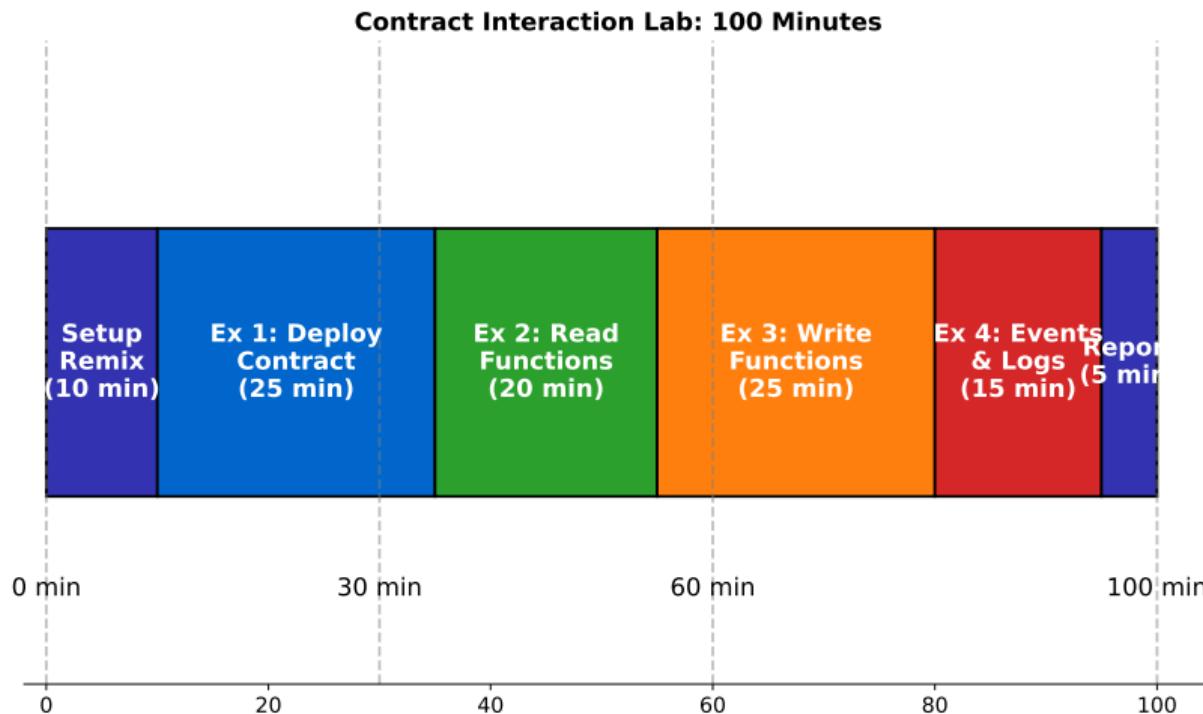
Course Instructor

Module B: Ethereum & Smart Contracts

By the end of this lab session, you will be able to:

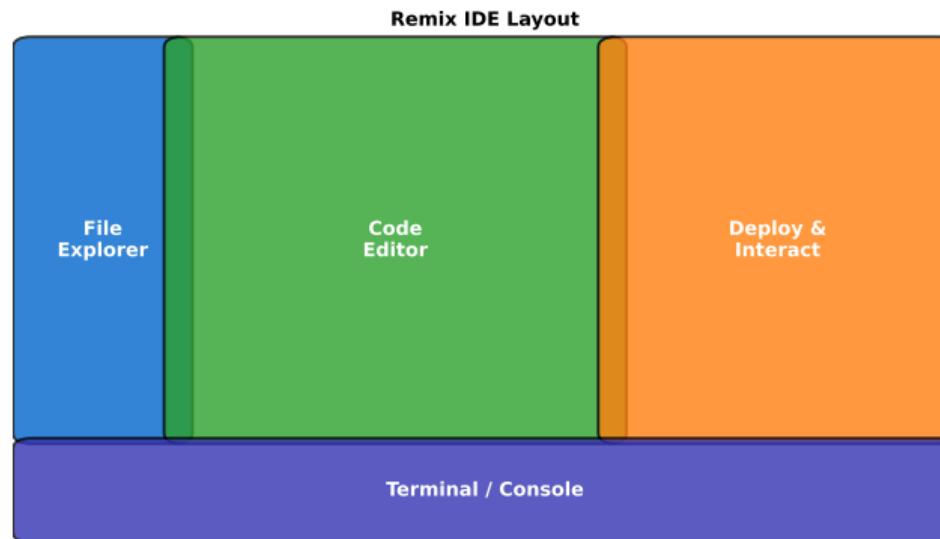
- Deploy smart contracts using Remix IDE
- Interact with contract read functions (view/pure)
- Execute contract write functions (state-changing)
- Monitor events and transaction logs
- Debug failed transactions

Lab Session Structure



Focus on hands-on interaction with deployed contracts

Remix IDE Interface



Deploy & Interact panel is where you'll spend most of your time

Exercise 1: Deploy a Simple Contract

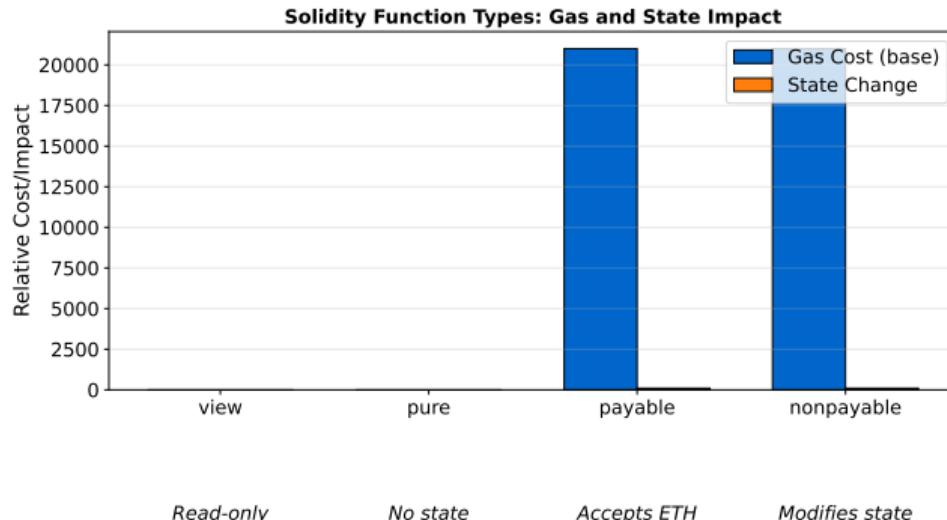
Steps:

- ① Open Remix IDE: <https://remix.ethereum.org>
- ② Create new file: SimpleStorage.sol
- ③ Write a storage contract with get/set functions
- ④ Compile using Solidity compiler
- ⑤ Deploy to JavaScript VM (local testing)

Key Concepts:

- Contract deployment creates new address
- Constructor runs once at deployment
- State variables persist on blockchain

Function Types in Solidity



View/pure functions are free; write functions cost gas

Exercise 2: Read Functions

Characteristics:

- **view**: Reads state, no modification
- **pure**: No state access, computation only
- No gas cost when called externally
- Instant response (no transaction needed)

Tasks:

- ① Call `get()` function to read stored value
- ② Observe: no MetaMask popup, instant result
- ③ Note the return value in Remix console

Exercise 3: Write Functions

Characteristics:

- Modify blockchain state
- Require gas payment
- Create transaction (needs confirmation)
- May emit events

Tasks:

- ① Call `set(42)` to store new value
- ② Observe: transaction created, gas consumed
- ③ Verify change by calling `get()`
- ④ Check transaction in Remix terminal

Exercise 4: Events and Logs

Why Events?

- Notify external applications of state changes
- Cheaper than storage for historical data
- Indexed parameters enable efficient filtering

Tasks:

- ① Add event to contract: `event ValueChanged(uint newValue)`
- ② Emit event in set function
- ③ Redeploy and test
- ④ View logs in Remix console

Submit:**① Solidity Contract:**

- SimpleStorage.sol with get, set, and event

② Screenshots:

- Deployed contract interface
- Read function result
- Write function transaction
- Event log output

Submission Deadline: One week from lab session

Key Takeaways

- Remix IDE is the standard tool for contract development
- View/pure functions are free (no gas)
- Write functions cost gas and create transactions
- Events provide efficient notification mechanism
- Always test on local VM before deploying to testnet