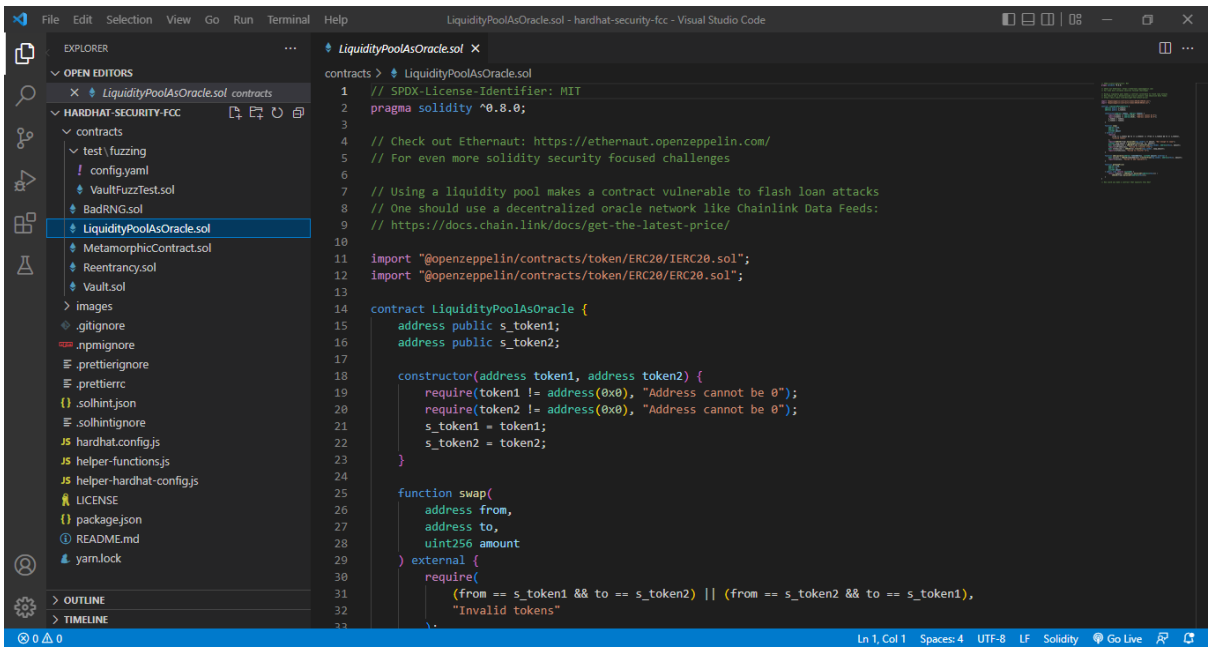
Lesson 18


File Edit Selection View Go Run Terminal Help
LiquidityPoolAsOracle.sol - hardhat-security-fcc - Visual Studio Code
EXPLORER
OPEN EDITORS
LiquidityPoolAsOracle.sol contracts
HARDHAT-SECURITY-FCC
contracts
test \ fuzzing
config.yaml
VaultFuzzTest.sol
BadRNG.sol
LiquidityPoolAsOracle.sol
MetamorphicContract.sol
Reentrancy.sol
Vault.sol
images
.gitignore
.npmignore
.prettierignore
.prettierrc
.solhint.json
.solhintignore
hardhat.config.js
helper-functions.js
helper-hardhat-config.js
LICENSE
package.json
README.md
yarn.lock
OUTLINE
TIMELINE
LiquidityPoolAsOracle.sol
contracts > LiquidityPoolAsOracle.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
// Check out Ethernaut: https://ethernaut.openzeppelin.com/
// For even more solidity security focused challenges
// Using a liquidity pool makes a contract vulnerable to flash loan attacks
// One should use a decentralized oracle network like Chainlink Data Feeds:
// https://docs.chain.link/docs/get-the-latest-price/
import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
contract LiquidityPoolAsOracle {
address public s_token1;
address public s_token2;
constructor(address token1, address token2) {
require(token1 != address(0), "Address cannot be 0");
require(token2 != address(0), "Address cannot be 0");
s_token1 = token1;
s_token2 = token2;
}
function swap(
address from,
address to,
uint256 amount
) external {
require(
(from == s_token1 && to == s_token2) || (from == s_token2 && to == s_token1),
"Invalid tokens"
);
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 LF Solidity Go Live