

Introduction to FinTech: CSIE5434 at NTU

Course website: [Must Read](#)

Homework 0 (“HW0”)

1 What is HW0 and why you makers should go do HW0 now:

- Read the course website shown on the title page first. Otherwise you won't know where to go pick up the course registration code later. Besides, all administrative updates will go to the website. I shall not update this latex document further, except if I notice a technical error. Again, course website is for non-technical, administrative business.
- While reading through this document, you need to click on all the underlined texts above and below, in order to learn about all the needed info that Professor Liao (liao@csie.ntu.edu.tw) provided for you.
- The purpose of HW0 is to facilitate the application process for the course. Please complete HW0 and provide your basic information for course enrollment. Note: After you enroll, a variant of HW0 will become HW1. So your endeavor NOW can still contribute to your final grade too, assuming you get a course enrollment code.
- HW0 has Problem 1 and 2. We observe that the earlier Web3 billionaires from this class (We made it!) are those who excelled in Problem 1. Those who fail Problem 1 are not hands-on. Instead of being makers, they are talkers and distract other students from making progress/money. They demand answers and spoon-feeding, instead of looking for answers themselves. So the purpose of Problem 1 is to find the makers and filter-out the talkers. In Web3, only makers can become money makers, sustainably. We are **FinTech**, not financial theoretician. Makers can't be cheated easily. We examine and set up github stuff carefully, as we also try to filter out cheaters in Problem 1. We don't want cheaters to enroll. HW0 serves the purpose to filter out those with the loser mentality.
- Enrolling involves the following 3 tasks:
 1. Finishing Problem 1 is a necessary but **NOT** a sufficient condition. Reason: At National Taiwan University we are limited by the size of classroom, but more than 250 students want to enroll. As Problem 1 only takes 30+ minutes, it will only serve as a low bar. You still need to complete the following tasks.
 2. Finishing Problem 2 (blockchain) will get you a higher priority of being enrolled. Doing smart contracts and Foundry (Problem 1) require blockchain knowledge. Waiver for this task of finishing Problem 2: By popular request, you can now show us your enrollment of CSIE5315 instead of answering Problem 2. The 2 earlier requests to me show that some want to get enrolled *right* after spending 30 minutes in doing Problem 1! Reason according to the requester: "I don't want to wait while the enrollment codes run out before I can finish Problem 2."
 3. Enrolling in the course "Bitcoin in the Big Data Era (CSIE5315)" will get you first priority, as that course covers the needed foundation in blockchain-related knowledge.
- CSIE5315 will be pre-recorded and you don't need to come to the morning class, as there won't be any quiz or roll calls.
- **Smart contract development** is foundational to Blockchain FinTech. Our course has a specific emphasis on EVM-based development of smart contracts.
 1. Solidity has been chosen as the programming language, and Foundry as our designated testing framework. Ensure that you can download and set up Foundry for this assignment.
 2. If you encounter difficulties with programming, it is advisable not to proceed with this course, as subsequent assignments will heavily involve exercises using this framework. There are many financial theory videos online. But CSIE5434 is science and real. CSIE5434 is offered each Spring, so you can come back when you become ready next year.
- The assignment comprises Problem 1 (a programming problem) and Problem 2 (3 research sub-problems).
- In case you encounter technical issues with Homework 0, kindly consult online resources initially, as most problems are likely related to your local environment. If challenges persist, reach out to 2b@csie.ntu.edu.tw following these guidelines:

1. Title your email [Fintech-HW0] [Summary-Of-Your-Issue]. Please note that we will **NOT** receive emails in other formats as we have applied filters to our email system.
2. Provide detailed information about your computer, including the operating system.
3. Outline the methods you attempted previously, the resources you consulted, the steps you followed, and the results of your efforts.
4. Note that ambiguous requests, such as attaching screenshots without proper descriptions, will not be answered. **Such emailing will lower your priority.**
5. Because you haven't enrolled yet, unfortunately we need to give priority to those who have enrolled. Professor's time is very limited, given 250 students. Most of problems can be solved on your own. Our SPEC is already very clear. Don't rush into Professor's office without working hard on your own first. Respect others' time!

2 Preliminaries

- [Git](#)
- [GitHub](#) Account
- [MetaMask](#) Wallet (or your preferred Web3 wallet supporting the Ethereum network)
- [Foundry](#)
 1. Install Foundry: `curl -L https://foundry.paradigm.xyz | bash`
 2. Install or update Foundry: `foundryup`
 3. Related Resource: [foundry book](#), [cheat sheet 1](#), [cheat sheet 2](#)
- Check forge version: `forge --version`
 1. The result will be similar to `forge 0.2.0 (c312c0d 2023-12-22T00:20:29.297186000Z)`

3 Problem 1

In this problem, we will download a default Foundry project - Counter, and introduce the Foundry project layout. Please follow the instructions below:

- Fork this [repository](#).
- Download this repository to your local environment.
- Enter the project directory `cd hw0` and run `forge install`
- Attempt to run all the tests: `forge test`, the result should look like [Figure 1](#)
- Get familiar with other forge command specified in the cheat sheet.

```
[#] Compiling...
[#] Compiling 1 files with 0.8.23
[#] Solc 0.8.23 finished in 1.39s
Compiler run successful!

Running 2 tests for test/Counter.t.sol:CounterTest
[PASS] testFuzz_SetNumber(uint256) (runs: 256, μ: 28020, ~: 28409)
[PASS] test_Increment() (gas: 28379)
Test result: ok. 2 passed; 0 failed; 0 skipped; finished in 23.41ms

Ran 1 test suites: 2 tests passed, 0 failed, 0 skipped (2 total tests)
```

Figure 1: forge test result

A Foundry project will have the following components:

- **src:** Contains the source code of your smart contract. Define and implement the functionality of your blockchain application.
- **test:** Test cases for your smart contract. This includes both unit tests and fuzz tests.
- **lib:** Include libraries and dependencies that your smart contract relies on.
- **script:** Include deployment scripts and automation scripts, or any other necessary scripts.

We will attempt to interact with the Ethereum test network - Sepolia, using Foundry scripts. Please follow the steps:

- Acquire some test ethers in the Sepolia network using the [faucet](#).
- Create a project on [Alchemy](#) and obtain the HTTPS endpoint from the API Key section. Read the [tutorial](#) for detailed guidance.
- Create a file named `.env` under `hw0` folder with two variables: `PRIVATE_KEY` and `SEPOLIA_RPC`.
 1. Ensure to include the `.env` file in the `.gitignore` file before pushing code to GitHub. Failure to do so may result in the exposure of your private key, leading to potential loss of funds!
- Fill the variables with the corresponding values (Note, add "0x" to your private key value if the prefix is not "0x").
 1. The result for the private key value will look like: `PRIVATE_KEY="0x...."`
 2. The result for the RPC endpoint value will look like: `SEPOLIA_RPC="⟨your-rpc-endpoint⟩"`
 3. Read the [tutorial](#) on retrieving the private key from your MetaMask account.

4. Exercise caution during the process, as leaking your private key can pose significant risks.
 5. Note that the private key is distinct from your wallet account address.
- Execute the command to deploy the contract, check [documentation](#) for more details.
`forge script script/Counter.s.sol:CounterScript --rpc-url $SEPOLIA_RPC --broadcast -vvv`
 - You will see a log value, and it will be your deployed contract address. The result will resemble the [Figure 2](#). Make sure to save the result as it is crucial for final submission.
 - Paste the address into Sepolia Etherscan, the result will resemble the [Figure 3](#).
 - Generate a GitHub token if you don't have one; it will be utilized for Git Actions later. Refer to the [documentation](#) for detailed guidance.
 - Push your code to GitHub. After pushing, navigate to the **Actions** section, where you should observe a pipeline labeled **Homework 0 Running by <Your-GitHub-Account>**. The result will resemble the [link](#).
 1. Avoid pushing to the official GitHub organization, **FintechIntro**.
 2. If the page displays the warning: **Workflows aren't being run on this forked repository**, Click the button with the caption: **I understand my workflows, go ahead and enable them**. Afterward, push the code again.
 3. Check the [link](#) and navigate to the **Run Forge Result** section. You will encounter a success message similar to the local one. This information will be utilized to verify your completion of homework in later assignments. You should encounter the same message in your CI/CD pipeline.
 - Congratulations! You have successfully completed all the steps in problem 1. Please capture screenshots similar to [Figure 2](#) and [Figure 3](#), then embed them in the README file within the repository.

```
[#] Compiling...
No files changed, compilation skipped
Script ran successfully.

== Logs ==
0xB4a76DBdb45Bdb2537161ce5a954aB22BD6f8DA4
```

Figure 2: The log value indicates the address of the deployed Counter contract.

The screenshot displays the Etherscan Sepolia Testnet interface. At the top, the search bar is set to 'Sepolia Testnet'. The main header shows the Etherscan logo and navigation links: Home, Blockchain, Tokens, NFTs, and Misc. Below the header, the contract address '0xB4a76DBdb45Bdb2537161ce5a954aB22BD6f8DA4' is highlighted. The 'Overview' section shows 'ETH BALANCE' as '0 ETH'. The 'More Info' section lists the 'CONTRACT CREATOR' as '0xd4c384...131f7057' at transaction '0x522049d1ed58e9c0...'. The 'Transactions' tab is active, showing a table with one transaction: '0x522049d1ed58e9c0...' with method '0xd0806040', block '5301819', age '1 day 14 hrs ago', from '0xd4c384...131f7057', to 'Contract Creation', value '0 ETH', and fee '0.00622555'. A footer note explains that a contract address hosts a smart contract.

Figure 3: Transaction success displayed on Etherscan in the Sepolia network.

4 Problem 2

There are various problems covering different topics throughout this course; please strive to address these problems with as much detail as possible. While you may refer to ChatGPT for guidance, ensure a thorough understanding of the following concept. Otherwise, later assignments might be quite challenging for you.

1. [Blockchain Basics] Explain the mechanism of gas on the blockchain, and outline the purpose of [EIP-1559](#), specifying the problem it addresses and solves.
2. [Smart Contract Basics] Review the Solidity [documentation](#) to enhance your understanding of storage layout, and illustrate the storage structure of the provided contract - [gist](#). In your report, provide the storage slot for each state variable (Preferred format: `var1 - slot 0, ...`). If a variable does not occupy any storage slot, please use `X`.
3. [DeFi Basics] Explain impermanent loss in Automated Market Maker (AMM), detailing the issue, and provide an illustrative example.

5 Submission

To turn in your homework, divide the content into two sections:

Part 1: Submission Instructions

- For Problem 1, include required screenshot in the README file of the repository, adhering to the specified policy.
- Document the solution for Problem 2 in a single file named `<Student-ID-HW0>` (Ex: B09902000) and attach it to the email.

Part 2: Email Content

- Subject: [Fintech-HW0] [Your-Name] [Your-Student-ID]. Please note that we will **NOT** receive emails in other formats as we have applied filters to our email system.
- Repository Link: Provide the link to your Homework 0 repository
- Course Enrollment or Domain Knowledge: Indicate whether you are enrolled in the course "Bitcoin in the Big Data Era (CSIE5315)" or provide evidence of your domain knowledge in blockchain or DeFi.

Part 3: Complete the [Google Spreadsheet](#) to streamline the process.

Please send the email to 2b@csie.ntu.edu.tw; late submissions will not be considered. The course code will be provided upon verification of the assignment.

6 History

Date	Version	Description
2024-02-19	1.0	Initial release