THE Hashgraph ASSOCIATION

HEDERA Certification

<Date>

THE Hashgraph ASSOCIATION

Certification

Duration: 3 Hours

Start: 10:00 CET (14:30 IST)

End: 13:00 CET (17:30 IST)

Location: Remote

SDK Language: Any

Open book (you can use any resources you need)

Ref: I1T4C3S2M5Z1

Task: Setup

Create a script to generate 5 Hedera Testnet accounts (Account1, Account2, Account3, Account4 and Account5).

Use these accounts as indicated in the following tasks.
These accounts must not already exist before the start of the test.

Be sure to note down the account Id and keys of the accounts – you will need these later. You will also need to communicate these account IDs during your response to the certification test.

Fund the accounts as you see appropriate to cover the costs of your tasks.

Task: Token Service

Create a script that generates a fungible token that requires a KYC process to be completed.

Set a fixed supply of 1000. Associate **Account3**, but do not KYC it. Try to send 12.99 tokens from **Account2** to **Account3**.

Show that the account is not yet able to participate in the token because it is not been KYC approved.

Now set the KYC flag on Account3 and retry the transfer.

Task: Smart Contract Service

Compile the Solidity contract given below using **Account1** and deploy it to the Hedera Network using the HCS with the ContractCreateFlow method. Write a script to call “function1” with the parameters 5 and 6. Gather the result and display it in the output.

Delete the smart contract.

Bytecode/ABI JSON URL	https://drive.proton.me/urls/HJT4ND5790#NbIrnDy64Idx

Task: Scheduled Transaction

Create a script that creates a scheduled transaction of 2 hbar from **Account1** to **Account2.**

Make a second script that deletes the transaction.

Print out the schedule information along the way along with the proof that the transfer did not happen.

Try to execute the transaction and show that it does not work.

Task: Multi Signature

Create a scheduled transaction with a key list with 3 key (**Account1, Account2** and **Account3**) that requires 2 of the three keys.

Sign the transaction with **Account1**. Get the information of the transaction and show that it has not yet been executed.

Sign the transaction with Account2 and get the information again to show that it has been executed.

Task: Consensus Service

Create a script to create a consensus transaction on the Hedera Consensus Service using **Account1**. Write the **current time** in the message of the transaction and submit.

THE Hashgraph ASSOCIATION

Finished

Please submit your results and code, either as a publicly available Git repository or Zip file back to the email you received the validation tasks from.

We will let you know the results within one working day.

Thanks!