

## Department of Computer Engineering

Academic Term: July-November 2023

**Class :** B.E Computers-A Sem VII

**Subject:** Blockchain Technology Lab

**Subject Code:** CSDL7022

<b>Practical No:</b>	6
<b>Title:</b>	Blockchain platform ethereum using Geth
<b>Date of Performance:</b>	8-9-2023
<b>Date of Submission:</b>	8-9-2023
<b>Roll No:</b>	9427
<b>Name of the Student:</b>	Atharva Prashant Pawar

### Evaluation:

Sr. No	Rubrics	Grades
1	Time Line (2)	
2	Output (3)	
3	Code optimization (2)	
4	Post lab (3)	

**Signature of the Teacher :**

## Experiment No. 6

### Blockchain platform ethereum using Geth.

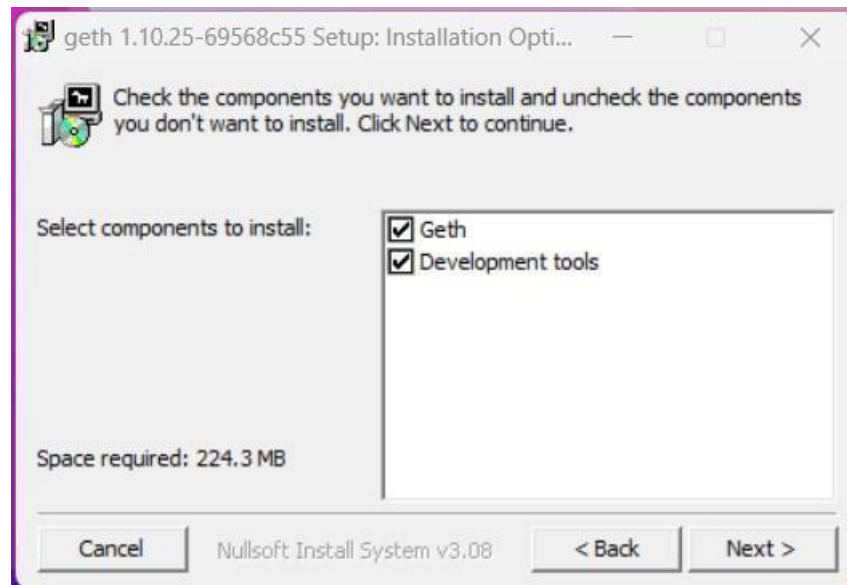
**Aim:** Study of Blockchain platform ethereum using Geth.

**Theory:**

**Step:1**

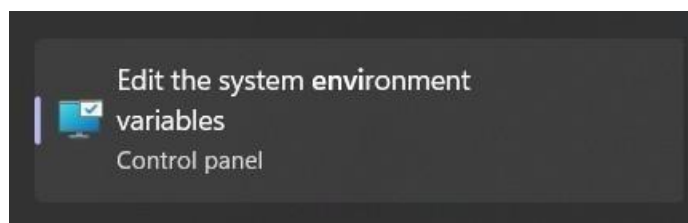
Following link is used to download Geth 1.10.25 for Windows.

<https://geth.ethereum.org/downloads/>

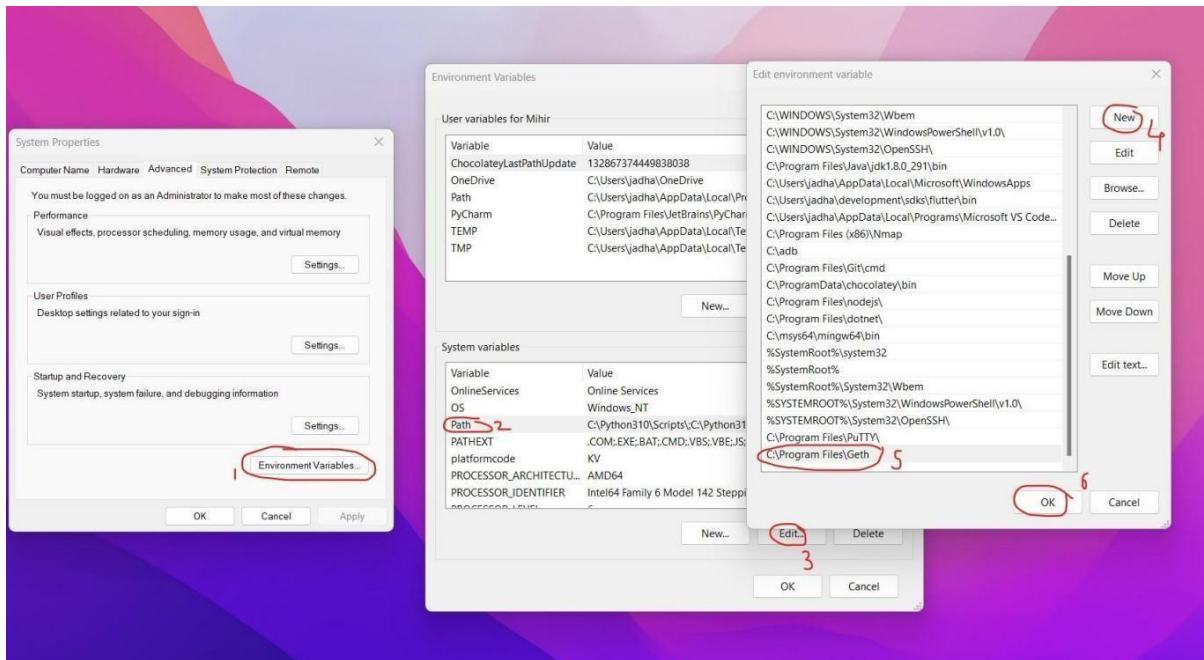


After installation add the Geth to the path

Search Environment Variables in Windows and Open it



After opening, add the Geth location i.e., C:\Program Files\Geth to the path in the Environment Variables



OR

Create folder in other drive say D drive and install Geth in that drive in folder Geth.

Following are the steps to be followed.

### Steps:

1. create a folder
2. create custom genesis file
3. create custom data directory
4. set custom networkID (ChainID)

## Step:2

### Create Genesis File

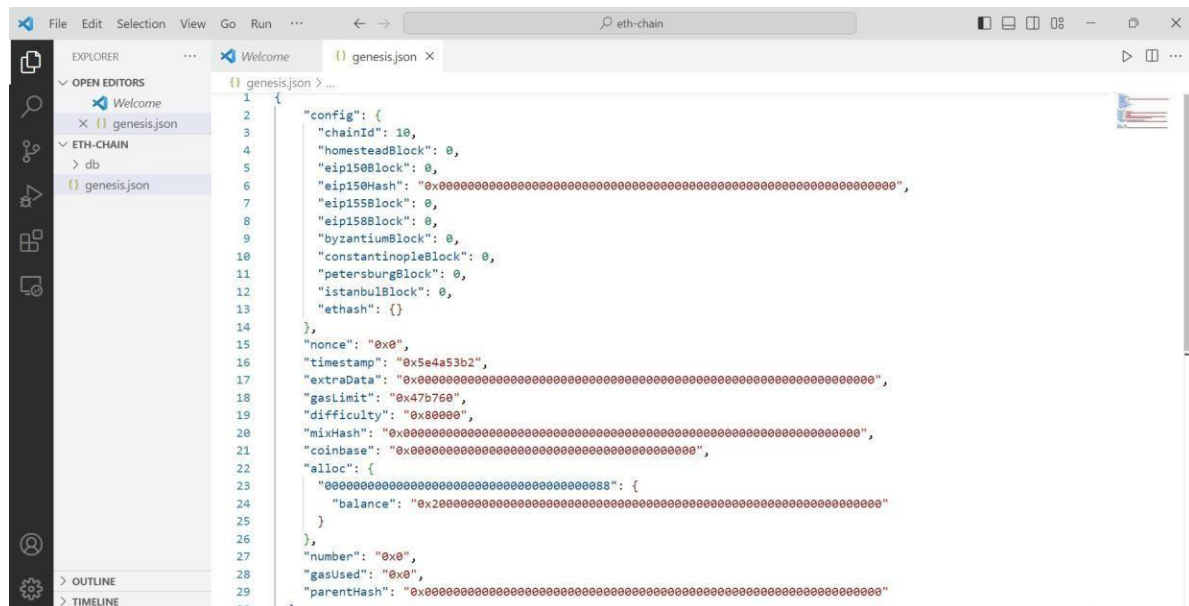
The Genesis block is the start block of the Blockchain — the first block, block 0, and the only block that does not point to a predecessor block. the genesis block is hard coded into clients, but in Ethereum it can be whatever you like. The Genesis file is a JSON file that defines the characteristics of that initial block and subsequently the rest of the blockchain.

1. Create a directory to hold your network files.

```
mkdir eth-chain
cd eth-chain
```

2. Create your genesis file

Open notepad and save file in eth-chain folder with name `genesis.json`. Copy following code in this `Genesis.json` file.

A screenshot of a code editor window titled 'eth-chain'. The Explorer sidebar on the left shows a project structure with 'OPEN EDITORS' containing 'Welcome' and 'genesis.json', and 'ETH-CHAIN' containing 'db' and 'genesis.json'. The main editor area displays the content of 'genesis.json' with line numbers 1 through 30. The JSON content is as follows:

```
1 {
2   "config": {
3     "chainId": 10,
4     "homesteadBlock": 0,
5     "eip150Block": 0,
6     "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
7     "eip155Block": 0,
8     "eip158Block": 0,
9     "byzantiumBlock": 0,
10    "constantinopleBlock": 0,
11    "petersburgBlock": 0,
12    "istanbulBlock": 0,
13    "ethash": {}
14  },
15  "nonce": "0x0",
16  "timestamp": "0x5e4a53b2",
17  "extraData": "0x0000000000000000000000000000000000000000000000000000000000000000",
18  "gasLimit": "0x47b760",
19  "difficulty": "0x80000",
20  "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
21  "coinbase": "0x0000000000000000000000000000000000000000000000000000000000000000",
22  "alloc": {
23    "0000000000000000000000000000000000000000000000000000000000000000": {
24      "balance": "0x2000000000000000000000000000000000000000000000000000000000000000"
25    }
26  },
27  "number": "0x0",
28  "gasUsed": "0x0",
29  "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000"
30 }
```

## config

`chainId` — this is your chain's identifier, and is used in replay protection.

`homesteadBlock`, `eip150Block`, `eip155Block`, `byzantiumBlock` — these relate to chain forking and versioning, so in our case let's leave them 0 since we're starting a new blockchain.

## difficulty

This dictates how difficult it is to mine a block.

## gasLimit

This is the total amount of gas that can be used in each block. With such a low mining difficulty, blocks will be moving pretty quick, but you should still set this value pretty high to avoid hitting the limit and slowing down your network.

## alloc

Here you can allocate ETH to specific addresses. This won't create the account for you, so make sure it's an account you already have control of. You will need to add the account to your private chain in order to use it, and to do that you need access to the keystore/utc file.

## Nonce

A scalar value equal to the number of transactions sent by the sender.

**mixhash**

mixhash is an intermediary calculation to finding the nonce that is not as costly to determine.

**coinbase**

The ether rewards gained from “mining” the genesis block go to the 160-bit coinbaseaddress.  
timestamp

The output of the Unix time() function when the block was created

**parentHash**

The Keccak 256-bit hash of the previous block’s header. This is meaningless in the genesis block, since block 0 has no parent.

**extraData**

An optional free, but max. 32 byte long space to conserve smart things for eternity on the Blockchain.

**Step:3****3. Initial the genesis block**

Init our blockchain with the settings in the genesis file and define a folder for storing chaindata.

```
> geth --datadir "/db" init genesis.json
```

datadir : Data directory for the databases and keystore

init : initialize a new genesis block

We get the following output’

```
Command Prompt - geth --d x Windows PowerShell x Windows PowerShell x + v
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>cd
C:\Users\User
C:\Users\User>cd ..
C:\Users>cd ..
C:\>mkdir eth-chain
C:\>cd eth-chain

C:\eth-chain>geth --datadir ".\db" init genesis.json
INFO [09-06|09:49:01.822] Maximum peer count          ETH=50 LES=0 total=50
INFO [09-06|09:49:01.831] Set global gas cap          cap=50,000,000
INFO [09-06|09:49:01.832] Allocated cache and file handles database=C:\eth-chain\db\geth\chaindata cache=16.00MiB handles=16
INFO [09-06|09:49:01.894] Opened ancient database      database=C:\eth-chain\db\geth\chaindata\ancient\chain readonly=false
INFO [09-06|09:49:01.894] Writing custom genesis block
INFO [09-06|09:49:01.895] Persisted trie from memory database nodes=1 size=171.00B time=0s gcsize=0.00B gctime=0s livenodes=1 liveness=0.00B
INFO [09-06|09:49:01.898] Successfully wrote genesis state database=chaindata hash=743236..b31e94
INFO [09-06|09:49:01.899] Allocated cache and file handles database=C:\eth-chain\db\geth\lightchaindata cache=16.00MiB handle
INFO [09-06|09:49:01.950] Opened ancient database      database=C:\eth-chain\db\geth\lightchaindata\ancient\chain readonly=false
INFO [09-06|09:49:01.951] Writing custom genesis block
INFO [09-06|09:49:01.952] Persisted trie from memory database nodes=1 size=171.00B time="868.2µs" gcsize=0.00B gctime=0s livenodes=1 liveness=0.00B
INFO [09-06|09:49:01.954] Successfully wrote genesis state database=lightchaindata hash=743236..b31e94
```

## Data Directory

Everything geth persists gets written inside its data directory (except for the PoW Ethash DAG. The default data directory locations in Window platform is %APPDATA%\Ethereum.

## Step:4

### Start your Ethereum peer node

Networkid helps ensure the privacy of your network. You can use any number here (where we used “ 123456”), but other peers joining your network must use the same one. Use following command.

```
geth --datadir ".\db" --networkid 123456 --http --http.port "8545" --http.corsdomain "*"
--nodiscover --http.api="admin,db,eth,debug,miner,net,shh,txpool,personal,web3"
```

```
Command Prompt - geth --d x Windows PowerShell X Windows PowerShell X + v
C:\eth-chain>geth --datadir ".\db" --networkid 123456 --http --http.port "8545" --http.corsdomain "*" --nodiscover --http.api "admin,
db,eth,debug,miner,net,shh,txpool,personal,web3"
INFO [09-06|09:50:07.613] Maximum peer count ETH=50 LES=0 total=50
INFO [09-06|09:50:07.622] Set global gas cap cap=50,000,000
INFO [09-06|09:50:07.623] Allocated trie memory caches clean=154.00MiB dirty=256.00MiB
INFO [09-06|09:50:07.624] Allocated cache and file handles database=C:\eth-chain\db\geth\chaindata cache=512.00MiB handles=81
92
INFO [09-06|09:50:07.656] Opened ancient database database=C:\eth-chain\db\geth\chaindata\ancient\chain readonly=fal
se
INFO [09-06|09:50:07.657]
INFO [09-06|09:50:07.657] -----
INFO [09-06|09:50:07.657] Chain ID: 10 (unknown)
INFO [09-06|09:50:07.657] Consensus: Ethash (proof-of-work)
INFO [09-06|09:50:07.657]
INFO [09-06|09:50:07.658] Pre-Merge hard forks:
INFO [09-06|09:50:07.658] - Homestead: 0 (https://github.com/ethereum/execution-specs/blob/master/network-u
pgrades/mainnet-upgrades/homestead.md)
INFO [09-06|09:50:07.658] - Tangerine Whistle (EIP 150): 0 (https://github.com/ethereum/execution-specs/blob/master/network-u
pgrades/mainnet-upgrades/tangerine-whistle.md)
INFO [09-06|09:50:07.658] - Spurious Dragon/1 (EIP 155): 0 (https://github.com/ethereum/execution-specs/blob/master/network-u
pgrades/mainnet-upgrades/spurious-dragon.md)
INFO [09-06|09:50:07.659] - Spurious Dragon/2 (EIP 158): 0 (https://github.com/ethereum/execution-specs/blob/master/network-u
pgrades/mainnet-upgrades/spurious-dragon.md)
INFO [09-06|09:50:07.659] - Byzantium: 0 (https://github.com/ethereum/execution-specs/blob/master/network-u
pgrades/mainnet-upgrades/byzantium.md)
INFO [09-06|09:50:07.659] - Constantinople: 0 (https://github.com/ethereum/execution-specs/blob/master/network-u
pgrades/mainnet-upgrades/constantinople.md)
INFO [09-06|09:50:07.659] - Petersburg: 0 (https://github.com/ethereum/execution-specs/blob/master/network-u
pgrades/mainnet-upgrades/petersburg.md)
INFO [09-06|09:50:07.659] - Istanbul: 0 (https://github.com/ethereum/execution-specs/blob/master/network-u
pgrades/mainnet-upgrades/istanbul.md)
```

```
Command Prompt - geth --d x Windows PowerShell X Windows PowerShell X + v
pgrades/mainnet-upgrades/istanbul.md)
INFO [09-06|09:50:07.659] - Berlin: <nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgr
ades/mainnet-upgrades/berlin.md)
INFO [09-06|09:50:07.660] - London: <nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgr
ades/mainnet-upgrades/london.md)
INFO [09-06|09:50:07.660]
INFO [09-06|09:50:07.660] The Merge is not yet available for this network!
INFO [09-06|09:50:07.660] - Hard-fork specification: https://github.com/ethereum/execution-specs/blob/master/network-upgrade
t-upgrades/paris.md
INFO [09-06|09:50:07.660] -----
INFO [09-06|09:50:07.660]
INFO [09-06|09:50:07.660] Disk storage enabled for ethash caches dir=C:\eth-chain\db\geth\ethash count=3
INFO [09-06|09:50:07.665] Disk storage enabled for ethash DAGs dir=C:\Users\User\AppData\Local\Ethash count=2
INFO [09-06|09:50:07.665] Initialising Ethereum protocol network=123,456 dbversion=<nil>
INFO [09-06|09:50:07.666] Loaded most recent local header number=0 hash=743236..b31e94 td=524,288 age=3y7mo6d
INFO [09-06|09:50:07.666] Loaded most recent local full block number=0 hash=743236..b31e94 td=524,288 age=3y7mo6d
INFO [09-06|09:50:07.666] Loaded most recent local fast block number=0 hash=743236..b31e94 td=524,288 age=3y7mo6d
WARN [09-06|09:50:07.667] Failed to load snapshot, regenerating err="missing or corrupted snapshot"
INFO [09-06|09:50:07.667] Rebuilding state snapshot root=5135f7..e3b389 accounts=0 slots=0 storage=0.00B dangling=0 el
apsed="611.3µs"
INFO [09-06|09:50:07.668] Generated state snapshot accounts=1 slots=0 storage=70.00B dangling=0 elapsed=1.178ms
INFO [09-06|09:50:07.669] Regenerated local transaction journal transactions=0 accounts=0
INFO [09-06|09:50:07.670] Gasprice oracle is ignoring threshold set threshold=2
WARN [09-06|09:50:07.670] Error reading unclean shutdown markers error="leveldb: not found"
WARN [09-06|09:50:07.670] Engine API enabled protocol=eth
WARN [09-06|09:50:07.670] Engine API started but chain not configured for merge yet
INFO [09-06|09:50:07.671] Starting peer-to-peer node instance=Geth/v1.10.25-stable-69568c55/windows-amd64/go1.18.5
INFO [09-06|09:50:07.690] New local node record seq=1,693,974,007,688 id=b7dd26ce8033dc7a ip=127.0.0.1 udp=0 tcp=3
0303
INFO [09-06|09:50:07.690] Started P2P networking self="enode://68244618a12c300f287055cd9cd93b8e73cf1208436d7e2ca8c"
```

## Step:5

Open a new command prompt. Use following command.

**geth attach http://127.0.0.1:8545**







```
Command Prompt - geth --di X Windows PowerShell X Windows PowerShell X + v
> admin.nodeInfo
{
  enode: "enode://68244618a12c300f287055cd9cdf93b8e73cf1208436d7e2ca8c68c8f4e1b696bc6ef08a671c6f9b0a7b54a883c644ff81838cf2e305390aaae
f1c40c483fd630127.0.0.1:30303?discport=0",
  enr: "-Jy4QJ63bzzr88ileHA4R5I9GLM3QpIaILDtNhMDzjpQL3Sx7dg5Qizb701iBjm8ECn32n2005Bto3ddP0ecpig_T2imGAYpouA-Ig2V0aMfGhFED95KAgmLkg
nY0gmLwhH8AAAGJc2VjcDI1NmsxoQNoJEYYoSwwDyhwVc2c35045zzxIIQ21-LKjGjI90G2LoRzbmFwwIN0Y3CCdL8",
  id: "b7dd26ce8033dc7ae5bf689e6954359b58f340ee187031df7ca93bc2d4f6e280",
  ip: "127.0.0.1",
  listenAddr: "[::]:30303",
  name: "Geth/v1.10.25-stable-69568c55/windows-amd64/go1.18.5",
  ports: {
    discovery: 0,
    listener: 30303
  },
  protocols: {
    eth: {
      config: {
        byzantiumBlock: 0,
        chainId: 10,
        constantinopleBlock: 0,
        eip150Block: 0,
        eip150Hash: "0x0000000000000000000000000000000000000000000000000000000000000000",
        eip155Block: 0,
        eip158Block: 0,
        ethash: {},
        homesteadBlock: 0,
        istanbulBlock: 0,
        petersburgBlock: 0
      },
      difficulty: 524288,
      genesis: "0x7432361e1877bc7fd490b7d7031014c272a51859d0ca203e58d94b9cc2b31e94",
      head: "0x7432361e1877bc7fd490b7d7031014c272a51859d0ca203e58d94b9cc2b31e94"
    }
  }
}
```

## Step:7

### Creating a new account

> **personal.newAccount()**

Creates a new account and prints the address. On the console, use:

Enter your password and then it will display account address. In my case, I set password to “123456”

Second way to create a new account

> **personal.newAccount("123456")**

```
Command Prompt - geth --d... X Windows PowerShell X Windows PowerShell X + v
> personal.newAccount()
Passphrase:
Repeat passphrase:
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.accounts
["0xa411af84b288b934b2322e437d531099627dd9f0"]
> eth.accounts[0]
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.accounts[0])
0
> eth.coinbase
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.coinbase)
0
> miner.start()
null
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
Error: contract creation without any data provided
    at web3.js:6365:9(45)
    at send (web3.js:5099:62(34))
    at <eval>:1:20(21)
> eth.getBalance(eth.coinbase)
```

## Step:8

### Check Accounts

Use following command to check account

> **eth.accounts**

```
Command Prompt - geth --d... X Windows PowerShell X Windows PowerShell X + v
Passphrase:
Repeat passphrase:
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.accounts
["0xa411af84b288b934b2322e437d531099627dd9f0"]
> eth.accounts[0]
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.accounts[0])
0
> eth.coinbase
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.coinbase)
0
> miner.start()
null
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
Error: contract creation without any data provided
    at web3.js:6365:9(45)
    at send (web3.js:5099:62(34))
    at <eval>:1:20(21)
> eth.getBalance(eth.coinbase)
0
> eth.getBalance(eth.coinbase)
```

Accounts is an array so you can search account by index also.

> `eth.accounts[0]`

```
> eth.accounts
["0xc31ea4c1325dfb2616a44827fa7f009406916c48", "0xef857ef3673a157db58895c12d00e624ce15acc4"]
> eth.accounts[0]
"0xc31ea4c1325dfb2616a44827fa7f009406916c48"
>
```

```
Command Prompt - geth --di x  Windows PowerShell  Windows PowerShell
Passphrase:
Repeat passphrase:
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.accounts
["0xa411af84b288b934b2322e437d531099627dd9f0"]
> eth.accounts[0]
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.accounts[0])
0
> eth.coinbase
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.coinbase)
0
> miner.start()
null
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
Error: contract creation without any data provided
    at web3.js:6365:9(45)
    at send (web3.js:5099:62(34))
    at <eval>:1:20(21)
> eth.getBalance(eth.coinbase)
0
> eth.getBalance(eth.coinbase)
```

## Step:9

### Check balance of account

> `eth.getBalance(eth.accounts[0])`

```
> eth.getBalance(eth.accounts[0])  
>  
>
```

```
Command Prompt - geth --datadir .\data X Windows PowerShell X Windows PowerShell X + v  
Passphrase:  
Repeat passphrase:  
"0xa411af84b288b934b2322e437d531099627dd9f0"  
> eth.accounts  
["0xa411af84b288b934b2322e437d531099627dd9f0"]  
> eth.accounts[0]  
"0xa411af84b288b934b2322e437d531099627dd9f0"  
> eth.getBalance(eth.accounts[0])  
0  
> eth.coinbase  
"0xa411af84b288b934b2322e437d531099627dd9f0"  
> eth.getBalance(eth.coinbase)  
0  
> miner.start()  
null  
> miner.stop()  
null  
> eth.getBalance(eth.coinbase)  
0  
> miner.stop()  
null  
> eth.getBalance(eth.coinbase)  
0  
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})  
Error: contract creation without any data provided  
    at web3.js:6365:9(45)  
    at send (web3.js:5099:62(34))  
    at <eval>:1:20(21)  
> eth.getBalance(eth.coinbase)  
0  
> eth.getBalance(eth.coinbase)|
```

## Step:10

## Mining

### Set Default Account

- Check your default account, type  
> `eth.coinbase`

```
Command Prompt - geth --d: X Windows PowerShell X Windows PowerShell X + v
Passphrase:
Repeat passphrase:
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.accounts
["0xa411af84b288b934b2322e437d531099627dd9f0"]
> eth.accounts[0]
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.accounts[0])
0
> eth.coinbase
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.coinbase)
0
> miner.start()
null
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
Error: contract creation without any data provided
    at web3.js:6365:9(45)
    at send (web3.js:5099:62(34))
    at <eval>:1:20(21)

> eth.getBalance(eth.coinbase)
0
> eth.getBalance(eth.coinbase)
```

## Start Mining

- Check your balance with

> **eth.getBalance(eth.coinbase)**

```
Command Prompt - geth --d: X Windows PowerShell X Windows PowerShell X + v
Passphrase:
Repeat passphrase:
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.accounts
["0xa411af84b288b934b2322e437d531099627dd9f0"]
> eth.accounts[0]
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.accounts[0])
0
> eth.coinbase
"0xa411af84b288b934b2322e437d531099627dd9f0"
> eth.getBalance(eth.coinbase)
0
> miner.start()
null
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> miner.stop()
null
> eth.getBalance(eth.coinbase)
0
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
Error: contract creation without any data provided
    at web3.js:6365:9(45)
    at send (web3.js:5099:62(34))
    at <eval>:1:20(21)

> eth.getBalance(eth.coinbase)
0
> eth.getBalance(eth.coinbase)
```

- Run

> `miner.start()`

```
> miner.start()
null
```

- Now, Mining will get started on the Blockchain network i.e., on the first Command Prompt
- Look at your other terminal window, you should see some mining action in the logs.

```
Command Prompt - geth --datadir "C:\mihirBlockchain" --networkid 123456 --http --http.port "8545" --http.corsdomain "*" --port 30303 --nodiscover...
INFO [09-23|19:27:24.707] mined potential block           number=9  hash=65d4e7..d14da4
INFO [09-23|19:27:24.707] Commit new sealing work         number=10 sealhash=f26a1d..376d6f uncles=0 txs=0 gas=
0 fees=0 elapsed=4.169s
INFO [09-23|19:27:24.710] Successfully sealed new block   number=10 sealhash=f26a1d..376d6f hash=5bc1e6..54b513
elapsed=4.172s
INFO [09-23|19:27:24.724] block reached canonical chain   number=3  hash=7c1b19..2c3d4f
INFO [09-23|19:27:24.716] Commit new sealing work         number=10 sealhash=f26a1d..376d6f uncles=0 txs=0 gas=
0 fees=0 elapsed=4.178s
INFO [09-23|19:27:24.727] mined potential block           number=10 hash=5bc1e6..54b513
INFO [09-23|19:27:24.749] Commit new sealing work         number=11 sealhash=19ef2a..92e4a1 uncles=0 txs=0 gas=
0 fees=0 elapsed=14.938ms
INFO [09-23|19:27:24.807] Commit new sealing work         number=11 sealhash=19ef2a..92e4a1 uncles=0 txs=0 gas=
0 fees=0 elapsed=73.457ms
INFO [09-23|19:27:25.117] Successfully sealed new block   number=11 sealhash=19ef2a..92e4a1 hash=c6fd36..f38707
elapsed=382.972ms
INFO [09-23|19:27:25.117] block reached canonical chain   number=4  hash=f491ae..09783a
INFO [09-23|19:27:25.141] Commit new sealing work         number=12 sealhash=14b22f..c55e9d uncles=0 txs=0 gas=
0 fees=0 elapsed=23.854ms
INFO [09-23|19:27:25.144] mined potential block           number=11 hash=c6fd36..f38707
INFO [09-23|19:27:25.259] Commit new sealing work         number=12 sealhash=14b22f..c55e9d uncles=0 txs=0 gas=
0 fees=0 elapsed=142.269ms
INFO [09-23|19:27:25.408] Successfully sealed new block   number=12 sealhash=14b22f..c55e9d hash=6fe538..335a7c
elapsed=290.947ms
INFO [09-23|19:27:25.408] block reached canonical chain   number=5  hash=23626b..4af5f0
INFO [09-23|19:27:25.448] Commit new sealing work         number=13 sealhash=804346..03711c uncles=0 txs=0 gas=
0 fees=0 elapsed=40.376ms
INFO [09-23|19:27:25.457] mined potential block           number=12 hash=6fe538..335a7c
INFO [09-23|19:27:25.502] Commit new sealing work         number=13 sealhash=804346..03711c uncles=0 txs=0 gas=
0 fees=0 elapsed=94.675ms
```



```
Command Prompt - geth --d X Windows PowerShell X Windows PowerShell X + v
> eth.getBalance(eth.accounts[0])
0
> eth.accounts[0]
"0xa411af84b288b934b2322e437d531099627dd9f0"
> miner.start()
null
> miner.stop()
null
> eth.getBalance(eth.accounts[0])
2600000000000000000
> eth.getBalance(eth.coinbase)
2600000000000000000
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
Error: contract creation without any data provided
    at web3.js:6365:9(45)
    at send (web3.js:5099:62(34))
    at <eval>:1:20(21)
> personal.newAccount("67890")
"0x2a36cbe63e2e0a5d244db42e4733bd44468e63db"
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
Error: authentication needed: password or unlock
    at web3.js:6365:9(45)
    at send (web3.js:5099:62(34))
    at <eval>:1:20(21)
> eth.getBalance(eth.accounts[1])
0
> eth.accounts
["0xa411af84b288b934b2322e437d531099627dd9f0", "0x2a36cbe63e2e0a5d244db42e4733bd44468e63db"]
>
```

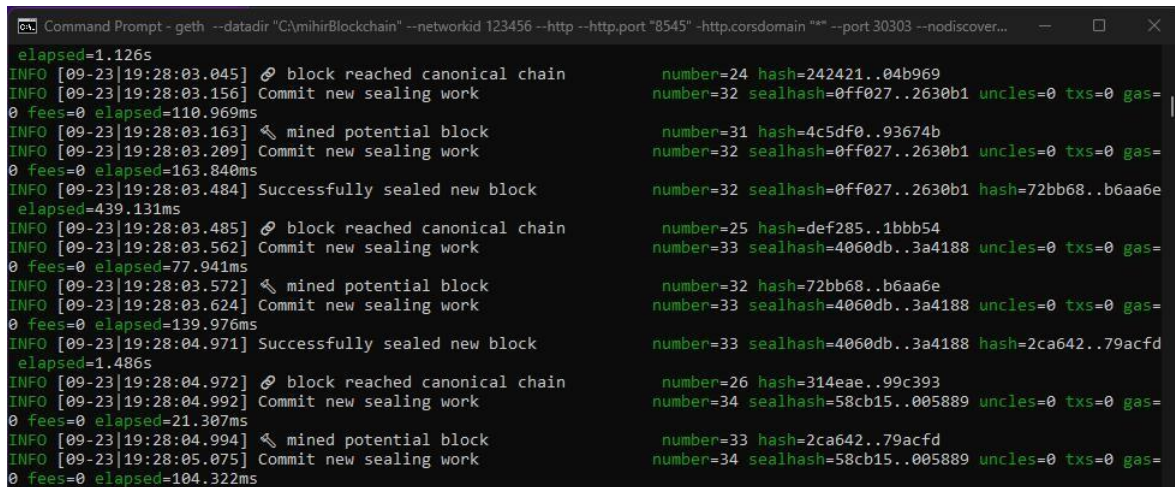
```
Command Prompt - geth --d X Windows PowerShell X Windows PowerShell X + v
INFO [09-06|10:03:53.001] Updated mining threads threads=8
INFO [09-06|10:03:53.002] Transaction pool price threshold updated price=1,000,000,000
INFO [09-06|10:03:53.005] Commit new sealing work number=1 sealhash=e6a42d..414417 uncles=0 txs=0 gas=0 fees=0 elaps
ed=3.264ms
INFO [09-06|10:03:53.006] Commit new sealing work number=1 sealhash=e6a42d..414417 uncles=0 txs=0 gas=0 fees=0 elaps
ed=3.877ms
INFO [09-06|10:04:07.125] Successfully sealed new block number=1 sealhash=e6a42d..414417 hash=c7dd09..d4199c elapsed=14.12
0s
INFO [09-06|10:04:07.127] ⚡ mined potential block number=1 hash=c7dd09..d4199c
INFO [09-06|10:04:07.129] Commit new sealing work number=2 sealhash=484b95..3f174a uncles=0 txs=0 gas=0 fees=0 elaps
ed=3.278ms
INFO [09-06|10:04:07.132] Commit new sealing work number=2 sealhash=484b95..3f174a uncles=0 txs=0 gas=0 fees=0 elaps
ed=6.578ms
INFO [09-06|10:04:07.991] Successfully sealed new block number=2 sealhash=484b95..3f174a hash=032a0d..3e057a elapsed=862.3
24ms
INFO [09-06|10:04:07.991] ⚡ mined potential block number=2 hash=032a0d..3e057a
INFO [09-06|10:04:07.996] Commit new sealing work number=3 sealhash=fc2000..b31e25 uncles=0 txs=0 gas=0 fees=0 elaps
ed=5.488ms
INFO [09-06|10:04:07.997] Commit new sealing work number=3 sealhash=fc2000..b31e25 uncles=0 txs=0 gas=0 fees=0 elaps
ed=6.496ms
INFO [09-06|10:04:08.576] Successfully sealed new block number=3 sealhash=fc2000..b31e25 hash=bc84b2..9bf510 elapsed=579.6
16ms
INFO [09-06|10:04:08.576] ⚡ mined potential block number=3 hash=bc84b2..9bf510
INFO [09-06|10:04:08.576] Commit new sealing work number=4 sealhash=f87384..74e0f1 uncles=0 txs=0 gas=0 fees=0 elaps
ed=0s
INFO [09-06|10:04:08.577] Commit new sealing work number=4 sealhash=f87384..74e0f1 uncles=0 txs=0 gas=0 fees=0 elaps
ed="668.6µs"
INFO [09-06|10:04:08.985] Successfully sealed new block number=4 sealhash=f87384..74e0f1 hash=aad9e7..60ed50 elapsed=409.3
36ms
INFO [09-06|10:04:08.985] ⚡ mined potential block number=4 hash=aad9e7..60ed50
INFO [09-06|10:04:08.985] Commit new sealing work number=5 sealhash=4dd397..58c014 uncles=0 txs=0 gas=0 fees=0 elaps
ed=0s
```



- To end mining, type

> `miner.stop()`

```
> miner.stop()  
null
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt - geth --datadir 'C:\mihirBlockchain' --networkid 123456 --http --http.port '8545' --http.corsdomain '\*' --port 30303 --nodiscover...". The window displays a continuous stream of logs from the Geth mining process. The logs are split into two columns. The left column contains informational messages such as "block reached canonical chain", "Commit new sealing work", "mined potential block", and "Successfully sealed new block", each followed by a timestamp and a duration (e.g., "elapsed=1.126s"). The right column displays the details of the mined blocks, including the block number, hash, sealhash, uncles, txs, and gas. The logs show a sequence of blocks being mined and sealed, with the process continuing to run.

```
elapsed=1.126s  
INFO [09-23|19:28:03.045] block reached canonical chain  
INFO [09-23|19:28:03.156] Commit new sealing work  
0 fees=0 elapsed=110.969ms  
INFO [09-23|19:28:03.163] mined potential block  
INFO [09-23|19:28:03.209] Commit new sealing work  
0 fees=0 elapsed=163.840ms  
INFO [09-23|19:28:03.484] Successfully sealed new block  
elapsed=430.131ms  
INFO [09-23|19:28:03.485] block reached canonical chain  
INFO [09-23|19:28:03.562] Commit new sealing work  
0 fees=0 elapsed=77.941ms  
INFO [09-23|19:28:03.572] mined potential block  
INFO [09-23|19:28:03.624] Commit new sealing work  
0 fees=0 elapsed=139.976ms  
INFO [09-23|19:28:04.971] Successfully sealed new block  
elapsed=1.486s  
INFO [09-23|19:28:04.972] block reached canonical chain  
INFO [09-23|19:28:04.992] Commit new sealing work  
0 fees=0 elapsed=21.307ms  
INFO [09-23|19:28:04.994] mined potential block  
INFO [09-23|19:28:05.075] Commit new sealing work  
0 fees=0 elapsed=104.322ms  
  
number=24 hash=242421..04b969  
number=32 sealhash=0ff027..2630b1 uncles=0 txs=0 gas=  
number=31 hash=4c5df0..93674b  
number=32 sealhash=0ff027..2630b1 uncles=0 txs=0 gas=  
number=32 sealhash=0ff027..2630b1 hash=72bb68..b6aa6e  
number=25 hash=def285..1bbb54  
number=33 sealhash=4060db..3a4188 uncles=0 txs=0 gas=  
number=32 hash=72bb68..b6aa6e  
number=33 sealhash=4060db..3a4188 uncles=0 txs=0 gas=  
number=33 sealhash=4060db..3a4188 hash=2ca642..79acfd  
number=26 hash=314eae..99c393  
number=34 sealhash=58cb15..005889 uncles=0 txs=0 gas=  
number=33 hash=2ca642..79acfd  
number=34 sealhash=58cb15..005889 uncles=0 txs=0 gas=
```

```
Command Prompt - geth --d X Windows PowerShell X Windows PowerShell X + v
590ms
INFO [09-06|10:04:17.265] block reached canonical chain
INFO [09-06|10:04:17.265] mined potential block
INFO [09-06|10:04:17.265] Commit new sealing work
sed="995.9µs"
INFO [09-06|10:04:17.265] Commit new sealing work
sed=1.507ms
INFO [09-06|10:04:18.512] Successfully sealed new block
7s
INFO [09-06|10:04:18.512] block reached canonical chain
INFO [09-06|10:04:18.512] mined potential block
INFO [09-06|10:04:18.513] Commit new sealing work
sed="687.7µs"
INFO [09-06|10:04:18.514] Commit new sealing work
sed=1.197ms
INFO [09-06|10:04:19.017] Successfully sealed new block
783ms
INFO [09-06|10:04:19.018] block reached canonical chain
INFO [09-06|10:04:19.018] mined potential block
INFO [09-06|10:04:19.018] Commit new sealing work
sed="594.5µs"
INFO [09-06|10:04:19.024] Commit new sealing work
sed=6.419ms
WARN [09-06|10:05:03.927] Served eth_sendTransaction
without any data provided"
INFO [09-06|10:05:45.714] Your new key was generated
WARN [09-06|10:05:45.714] Please backup your key file!
-2a36cbe63e2e0a5d244db42e4733bd44468e63db
WARN [09-06|10:05:45.714] Please remember your password!
WARN [09-06|10:05:51.090] Served eth_sendTransaction
on needed: password or unlock"
number=4 hash=aad9e7..60ed50
number=11 hash=a832f7..406277
number=12 sealhash=d8fa4c..e3eff3 uncles=0 txs=0 gas=0 fees=0 elap
number=12 sealhash=d8fa4c..e3eff3 uncles=0 txs=0 gas=0 fees=0 elap
number=12 sealhash=d8fa4c..e3eff3 hash=467606..8053f7 elapsed=1.24
number=5 hash=17a31f..13ea75
number=12 hash=467606..8053f7
number=13 sealhash=6ace9d..1cdd30 uncles=0 txs=0 gas=0 fees=0 elap
number=13 sealhash=6ace9d..1cdd30 uncles=0 txs=0 gas=0 fees=0 elap
number=13 sealhash=6ace9d..1cdd30 hash=4dbac1..c7b471 elapsed=504.
number=6 hash=c151dd..d76842
number=13 hash=4dbac1..c7b471
number=14 sealhash=2b2f76..d90038 uncles=0 txs=0 gas=0 fees=0 elap
number=14 sealhash=2b2f76..d90038 uncles=0 txs=0 gas=0 fees=0 elap
conn=127.0.0.1:63309 reqid=22 duration="549.3µs" err="contract cre
address=0x2A36cBE63e2e0a5D244dB42E4733BD44468E63db
path=C:\eth-chain\db\keystore\UTC--2023-09-06T04-35-43.170007300Z-
conn=127.0.0.1:63309 reqid=26 duration=16.5015ms err="authenticati
```

## Transfer

## 1. Check your balance

```
> eth.getBalance(eth.coinbase)
```

```
0  
> eth.getBalance(eth.accounts[0])  
0  
> eth.accounts[0]  
"0xa411af84b288b934b2322e437d531099627dd9f0"  
> miner.start()  
null  
> miner.stop()  
null  
> eth.getBalance(eth.accounts[0])  
26000000000000000000  
> eth.getBalance(eth.coinbase)  
26000000000000000000  
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})  
Error: contract creation without any data provided  
    at web3.js:6365:9(45)  
    at send (web3.js:5099:62(34))  
    at <eval>:1:20(21)  
  
> personal.newAccount("67890")  
"0x2a36cbe63e2e0a5d244db42e4733bd44468e63db"  
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})  
Error: authentication needed: password or unlock  
    at web3.js:6365:9(45)  
    at send (web3.js:5099:62(34))  
    at <eval>:1:20(21)  
  
> eth.getBalance(eth.accounts[1])  
0  
> eth.accounts  
["0xa411af84b288b934b2322e437d531099627dd9f0", "0x2a36cbe63e2e0a5d244db42e4733bd44468e63db"]  
>
```

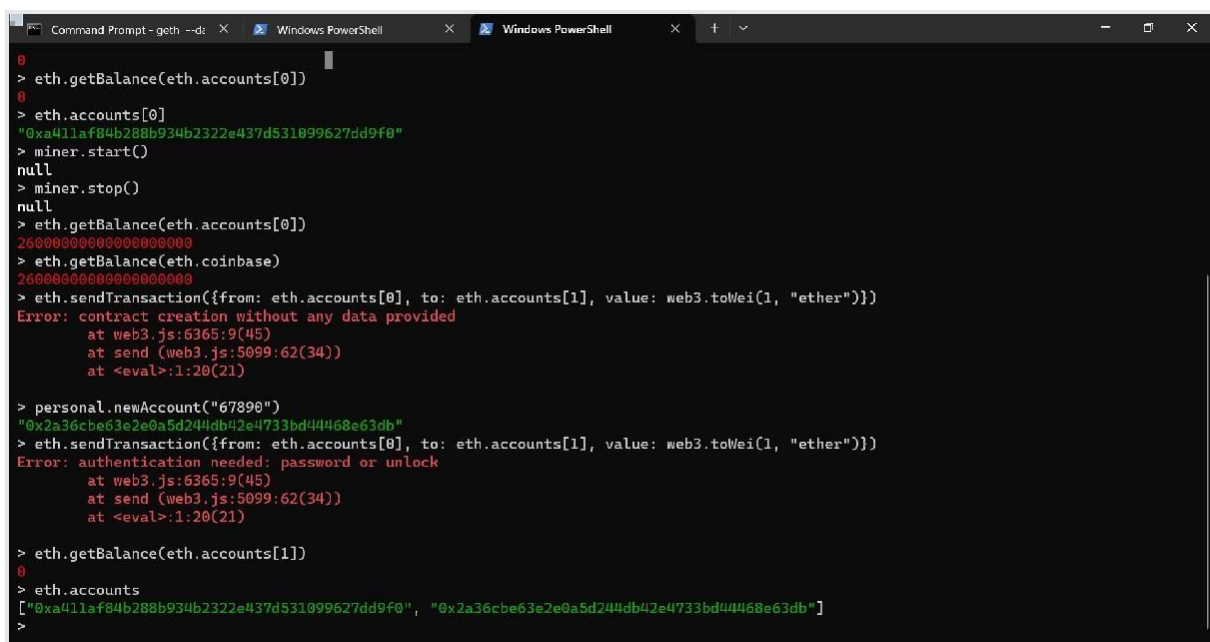
**2. Transfer Ether, type:**

```
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value:
web3.toWei(1, "ether")})
```

OR

```
> eth.sendTransaction({from: 'your_first_account_address', to:
'your_second_account_address', value: web3.toWei(1, "ether")})
```

But you should get error like this.

A screenshot of a Windows PowerShell terminal window with a dark background. The window title bar shows 'Command Prompt - geth --ds', 'Windows PowerShell', and 'Windows PowerShell'. The terminal displays a series of commands and their outputs. It starts with 'eth.getBalance(eth.accounts[0])' returning '0'. Then 'eth.accounts[0]' returns a hexadecimal address. 'miner.start()' returns 'null', and 'miner.stop()' also returns 'null'. 'eth.getBalance(eth.accounts[0])' returns '26000000000000000000'. 'eth.getBalance(eth.coinbase)' returns '26000000000000000000'. The next command is 'eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})', which results in a red error message: 'Error: contract creation without any data provided' with stack traces. Then 'personal.newAccount("67890")' returns a new address. The next 'eth.sendTransaction' command results in a red error: 'Error: authentication needed: password or unlock' with stack traces. Finally, 'eth.getBalance(eth.accounts[1])' returns '0', and 'eth.accounts' returns an array of two addresses.

```
> eth.getBalance(eth.accounts[0])
0
> eth.accounts[0]
"0xa411af84b288b934b2322e437d531099627dd9f0"
> miner.start()
null
> miner.stop()
null
> eth.getBalance(eth.accounts[0])
26000000000000000000
> eth.getBalance(eth.coinbase)
26000000000000000000
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
Error: contract creation without any data provided
    at web3.js:6365:9(45)
    at send (web3.js:5099:62(34))
    at <eval>:1:20(21)
> personal.newAccount("67890")
"0x2a36cbe63e2e0a5d244db42e4733bd44468e63db"
> eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
Error: authentication needed: password or unlock
    at web3.js:6365:9(45)
    at send (web3.js:5099:62(34))
    at <eval>:1:20(21)
> eth.getBalance(eth.accounts[1])
0
> eth.accounts
["0xa411af84b288b934b2322e437d531099627dd9f0", "0x2a36cbe63e2e0a5d244db42e4733bd44468e63db"]
>
```

Actually, User have to unlock account first before transfer.

**Conclusion:** We have succesfully installed Geth and performed all the commands for various operations.

Q. Observations:

- He first install geth which is an ethereum client meaning it handles transaction, deployment & execution of smart contracts & contains an embedded computing device known as EVM.
- He demonstrate how to do mining using geth.