**PRACTICAL 9**

**AIM:**

Build the private blockchain network using Geth.

**STEPS OF IMPLEMENTATION:**

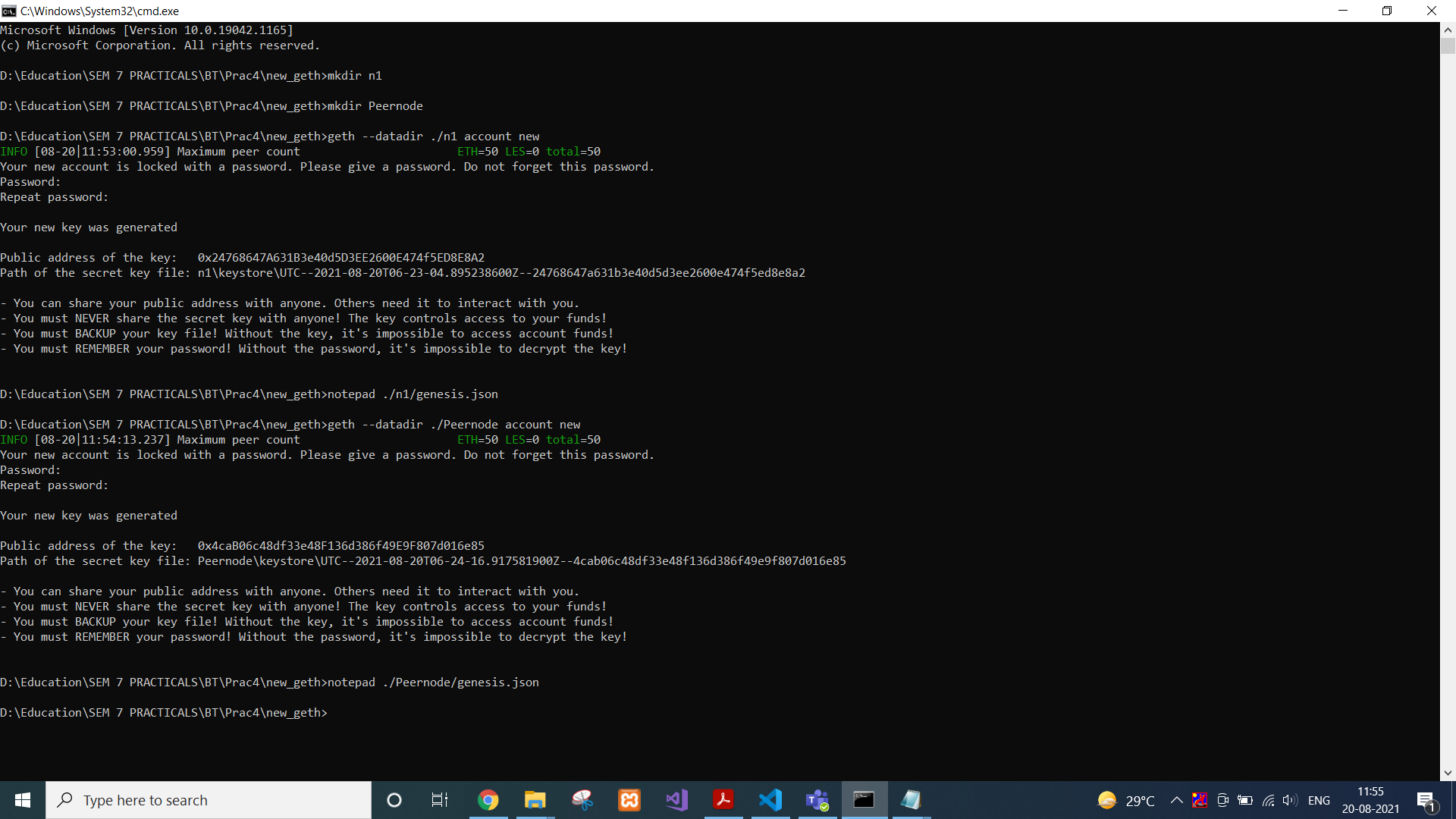
First we will create a main directory. Here we’re naming it “new\_geth” but any name will work. Then in cmd, we will navigate to it.

cd new\_geth

Then we will make 2 directories inside that folder. We’re naming then n1 and peernode.

mkdir n1

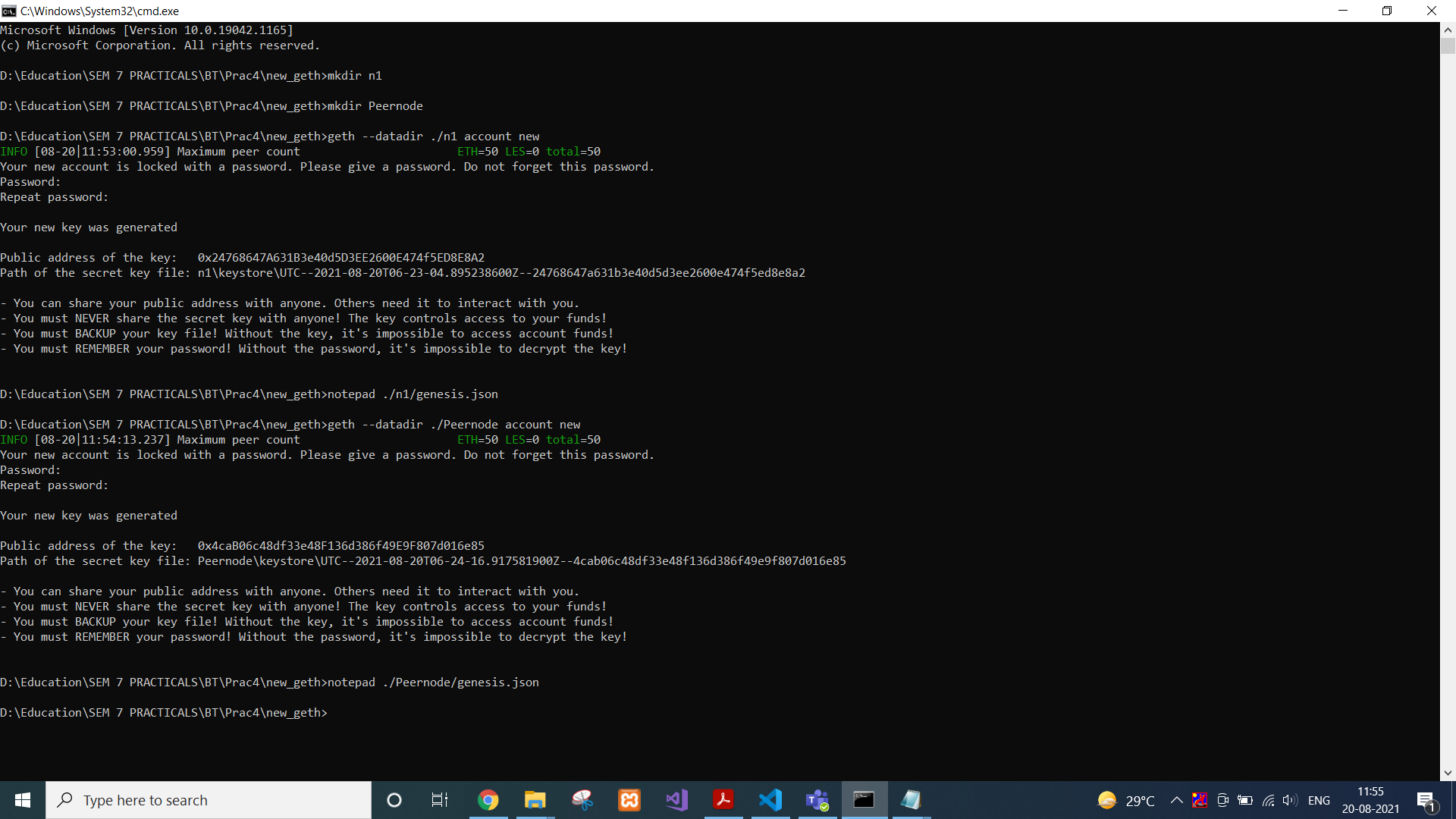
mkdir Peernode



Then we will create new accounts in both the folders with following commands.

geth --datadir ./n1 account new

geth --datadir ./Peernode account new



That would generate public address of both accounts.

We will create genesis.json file in both directories with following code.

{

"config": {

"chainId": 1234,

"homesteadBlock": 0,

"eip150Block": 0,

"eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",

"eip155Block": 0,

"eip158Block": 0,

"byzantiumBlock": 0

},

"difficulty": "400",

"gasLimit": "2000000",

"alloc": {

"<public-address-of-the-key>": {

"balance": "500000000000000000000000"

}

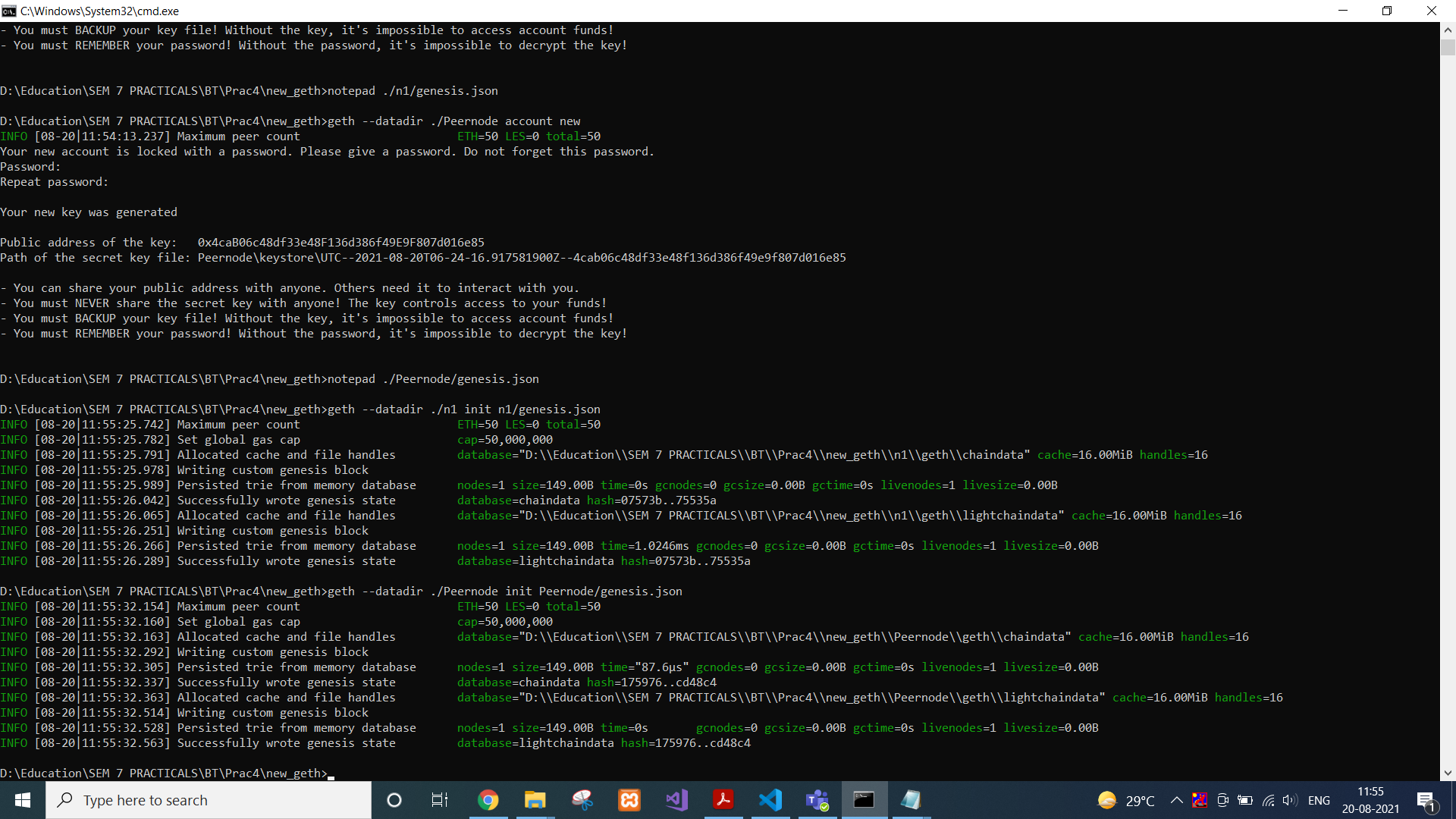
}

}

Then we will initiate both the genesis blocks.

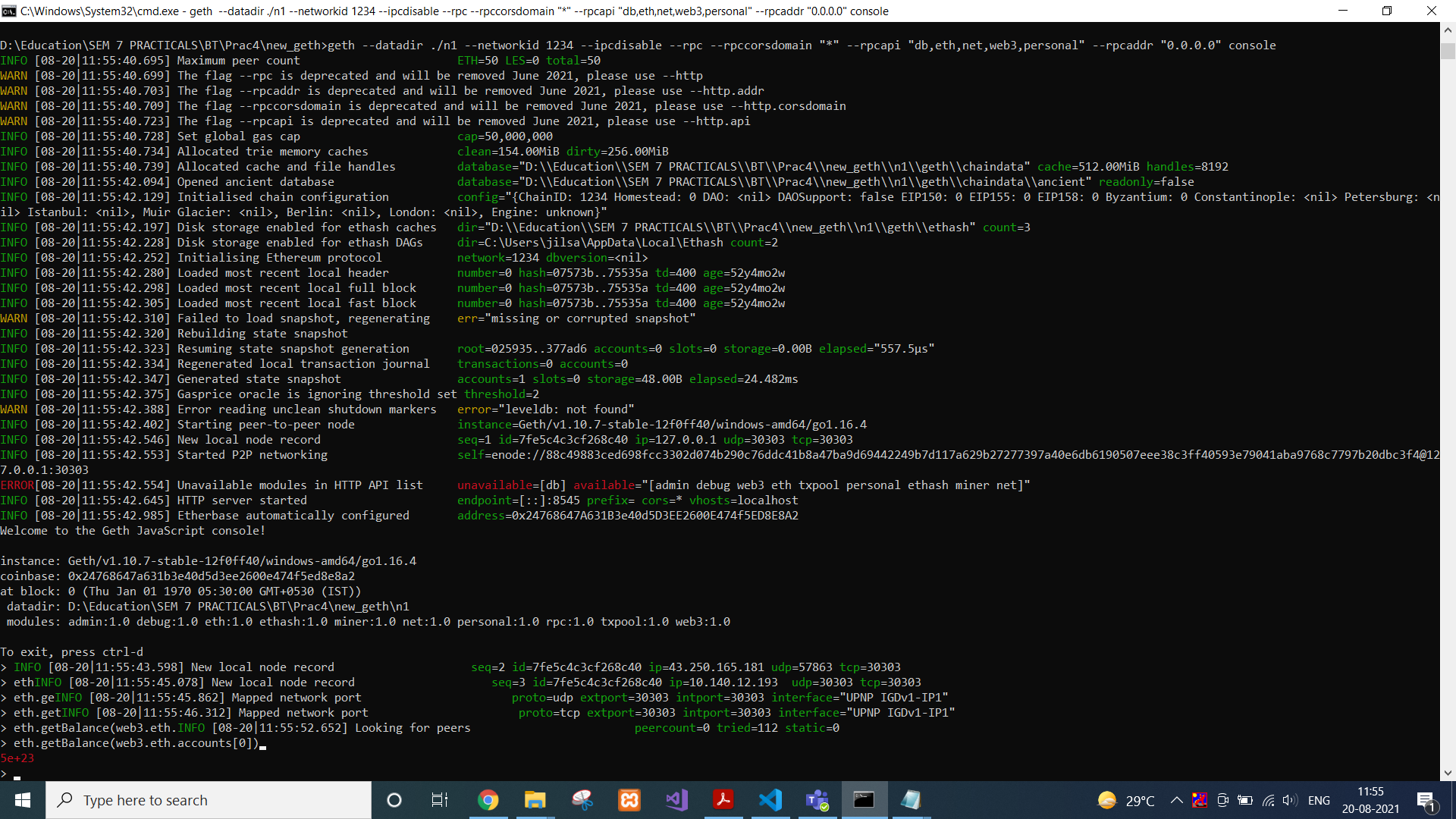
geth --datadir ./n1 init n1/genesis.json

geth --datadir ./Peernode init Peernode/genesis.json



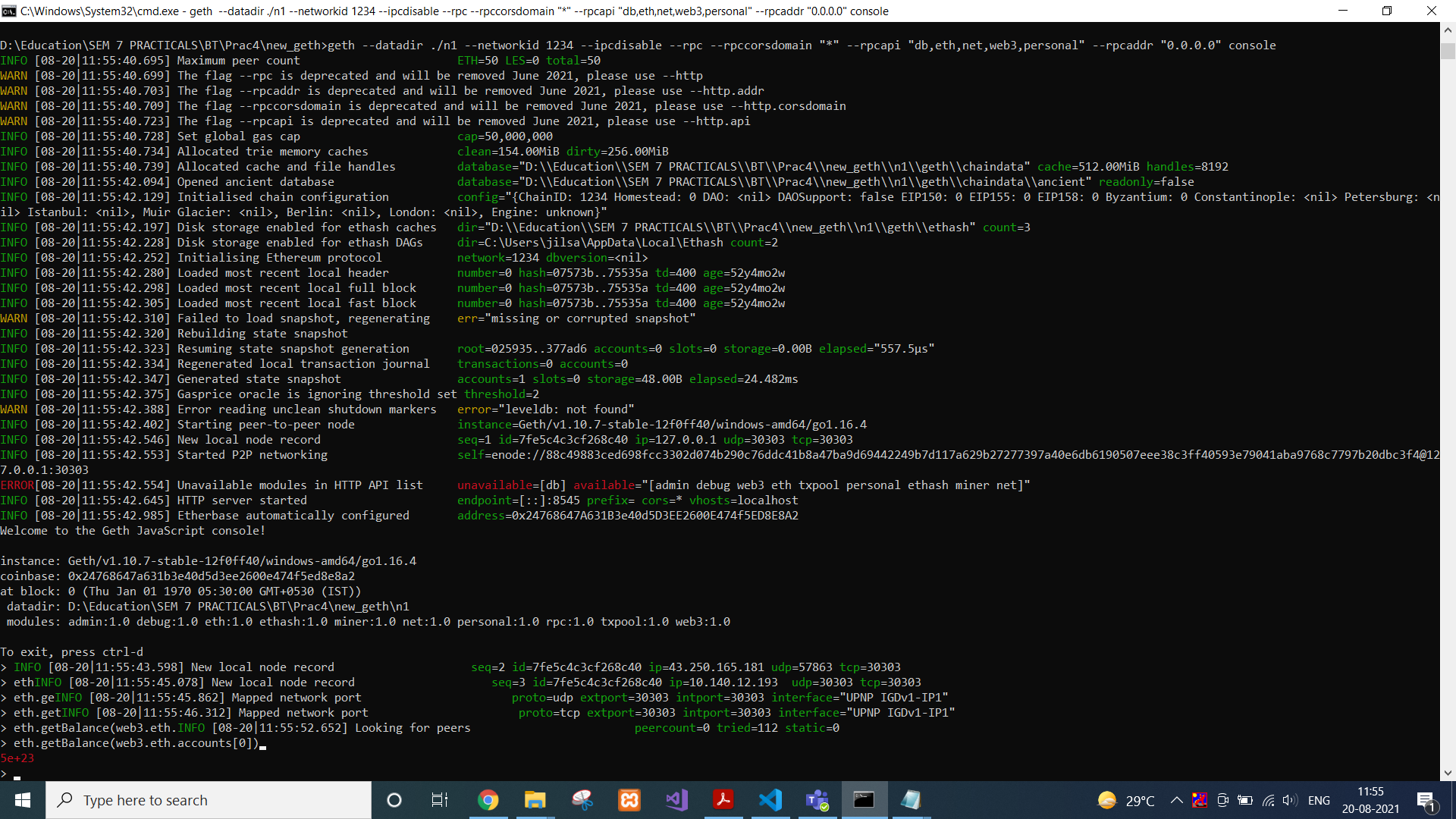
Then we will start interactive javascript environment to interact with our newly created blockchain.

geth --datadir ./n1 --networkid 1234 --ipcdisable --rpc --rpccorsdomain "\*" --rpcapi "db,eth,net,web3,personal" --rpcaddr "0.0.0.0" console



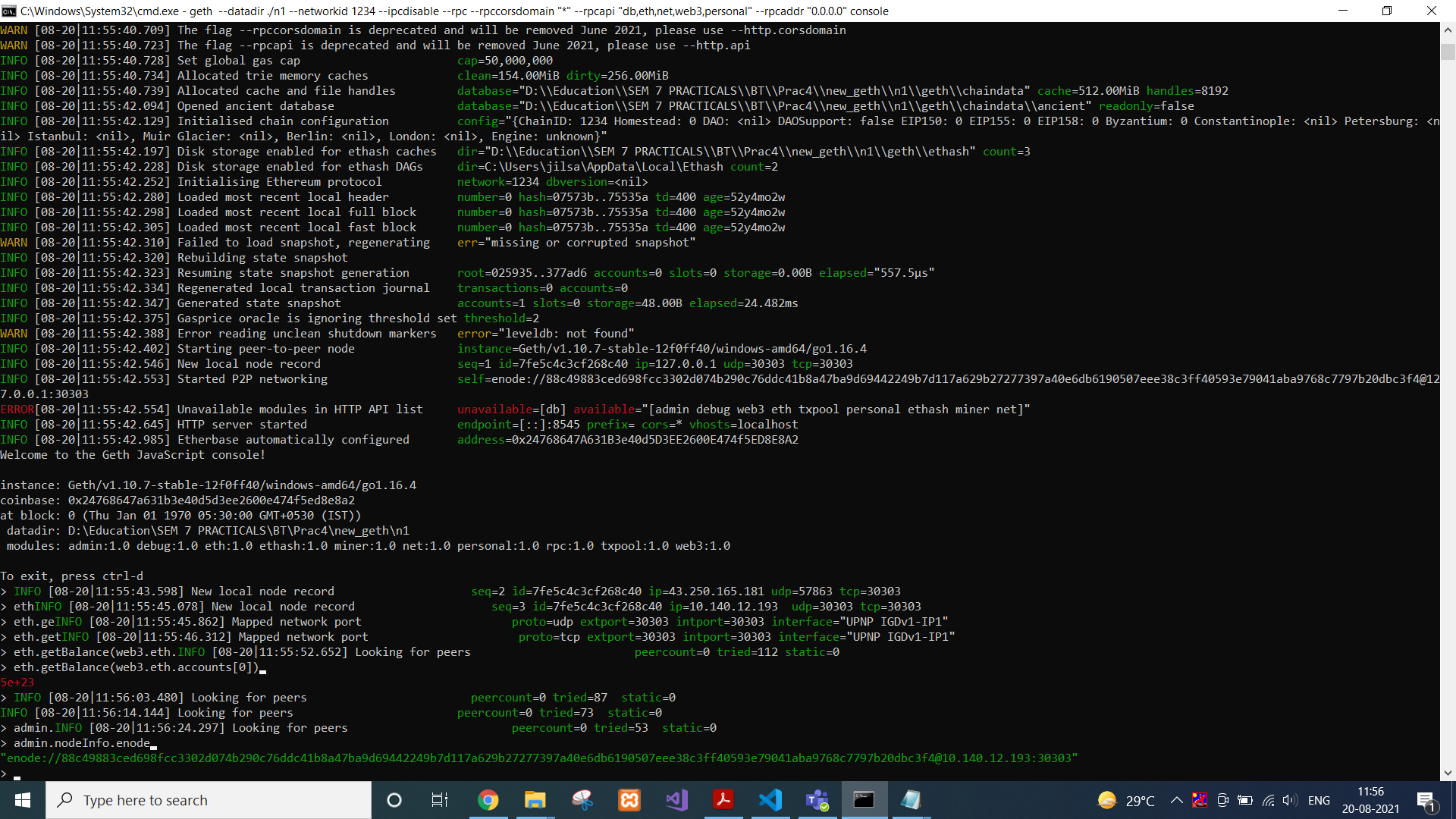
To check balance in our account, we will simply type the following command while the background process is still running.

eth.getBalance(web3.eth.accounts[0])



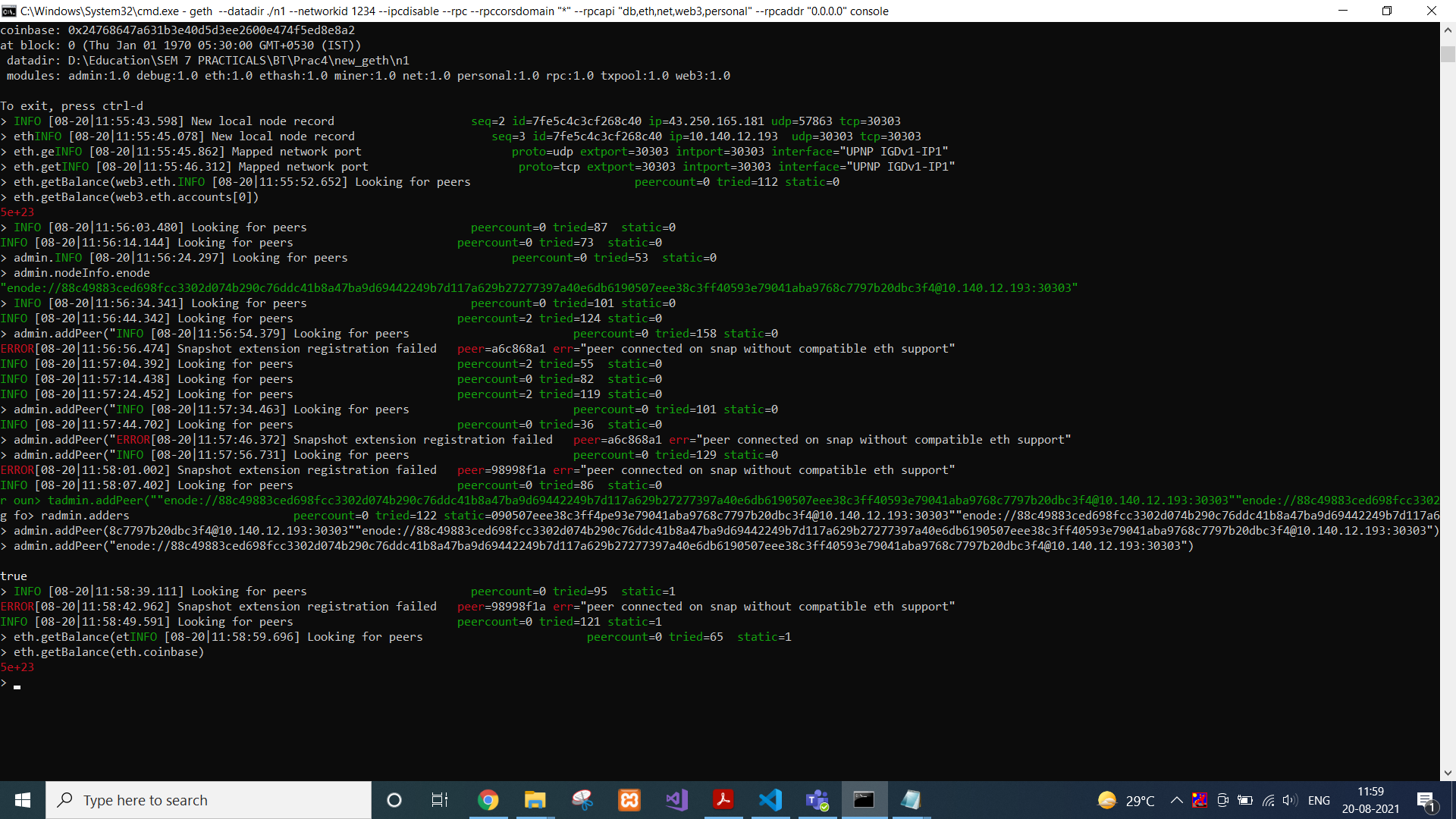
Admin object exposes methods to interact with the RPC APIs. To get information about enode, we will write following command.

admin.nodeInfo.enode



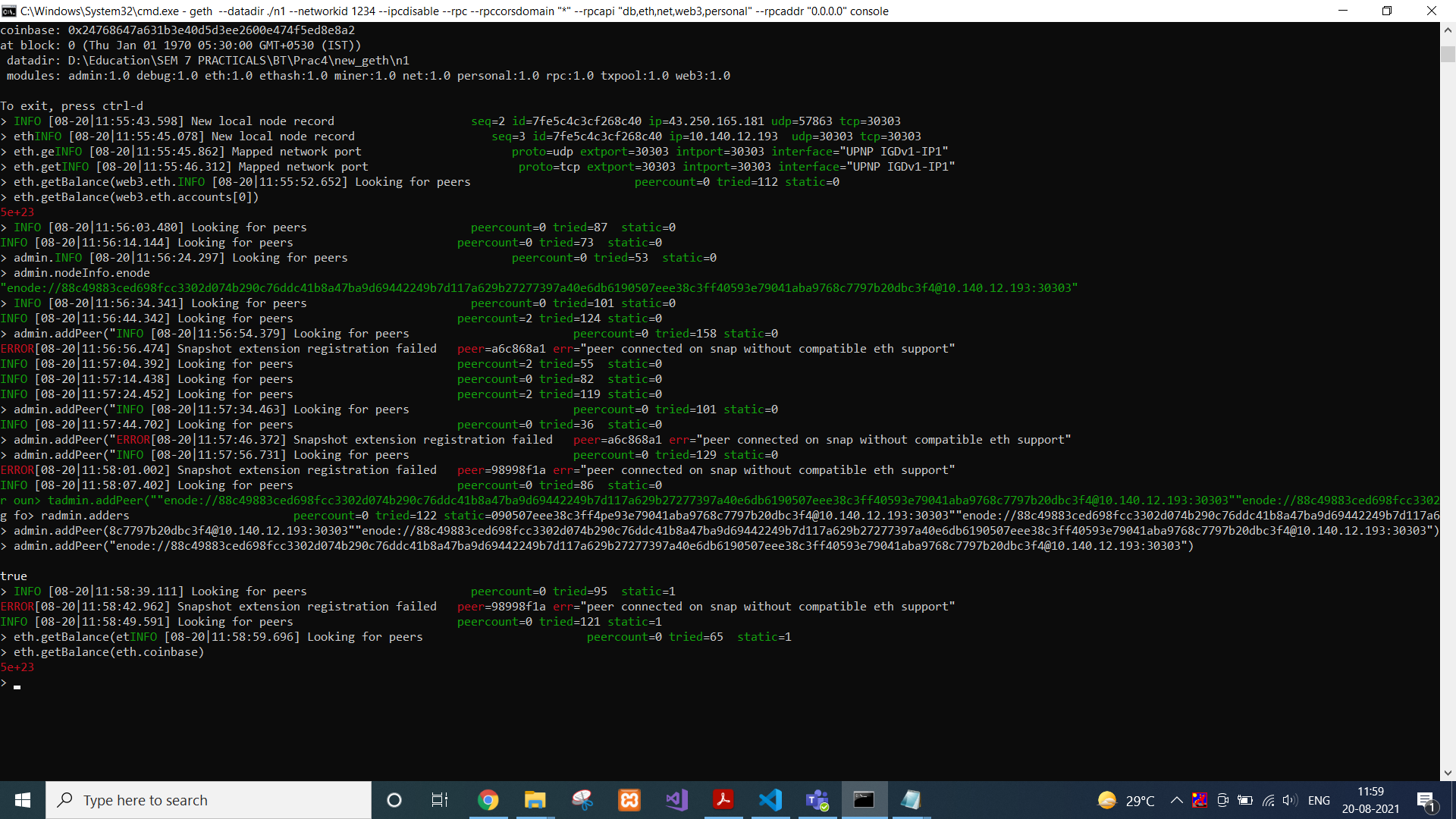
We will copy this enode address and Request adding a new remote node to the list of tracked static nodes.

admin.addPeer("paste the enode address")



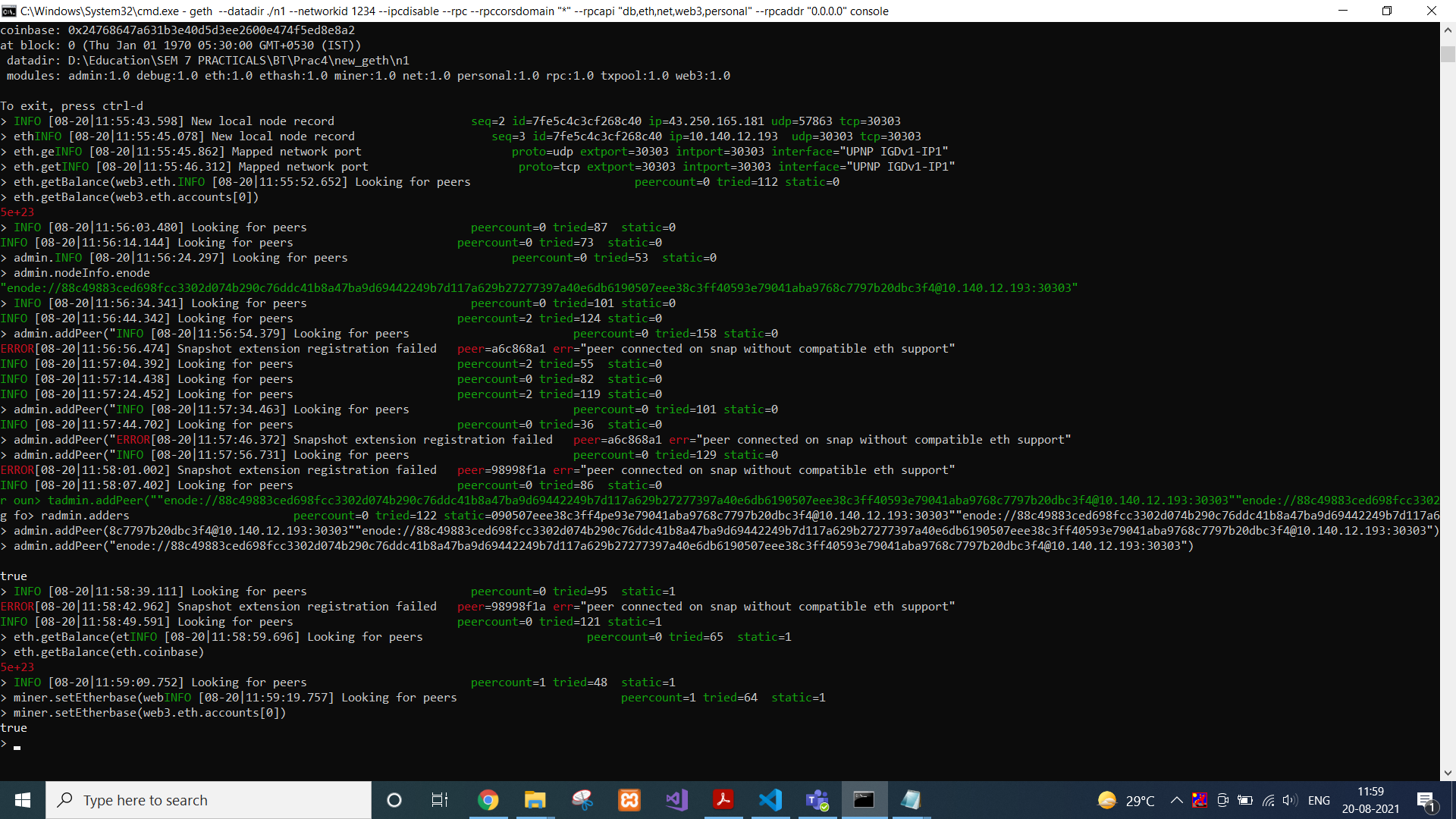
To check your the etherbase account balance, we will use…

eth.getBalance(eth.coinbase)



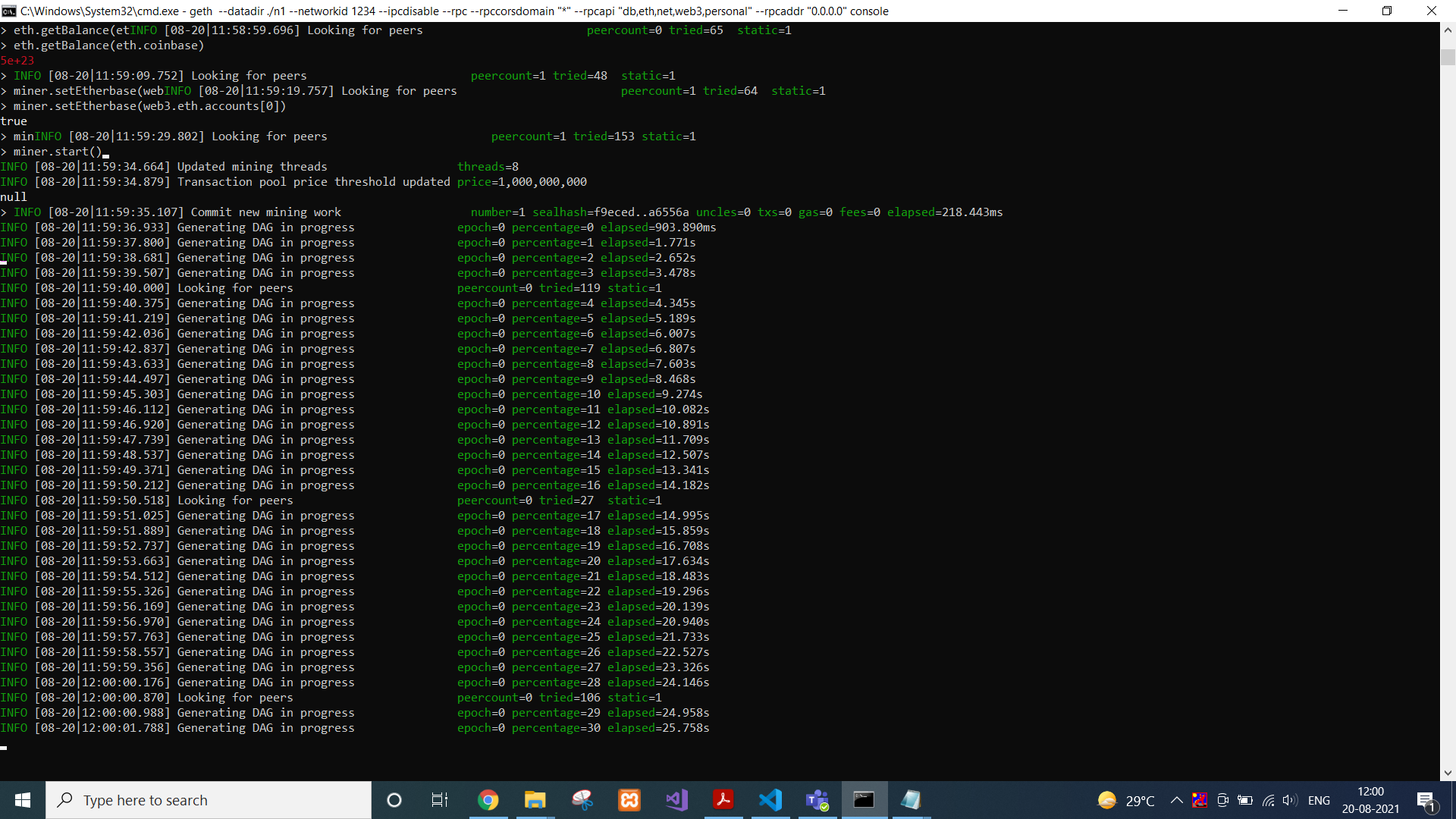
Now we will set the etherbase, where mining rewards will go.

miner.setEtherbase(web3.eth.accounts[0])



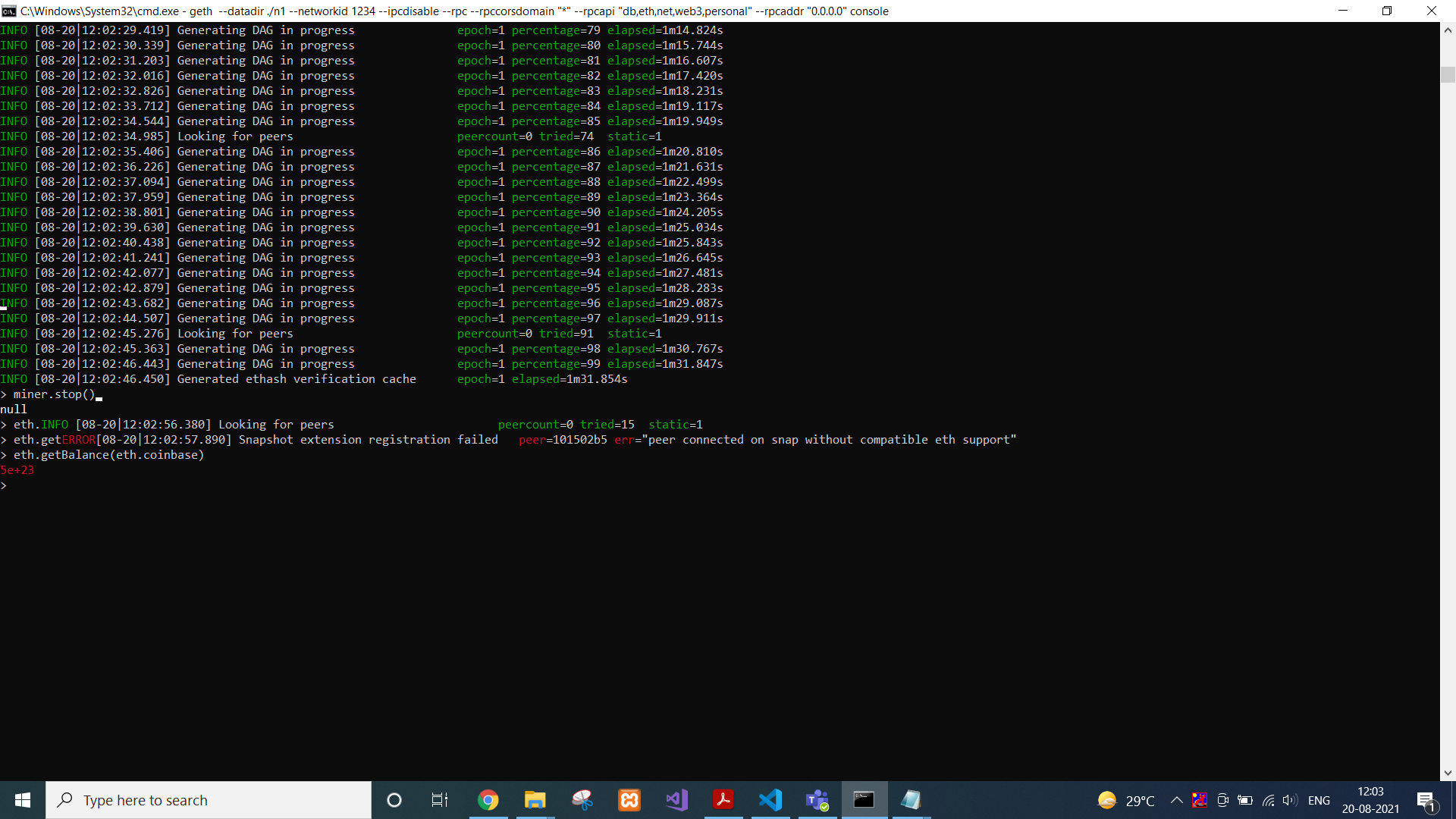
Then finally, we are ready to mine some Ethereum now.

miner.start()



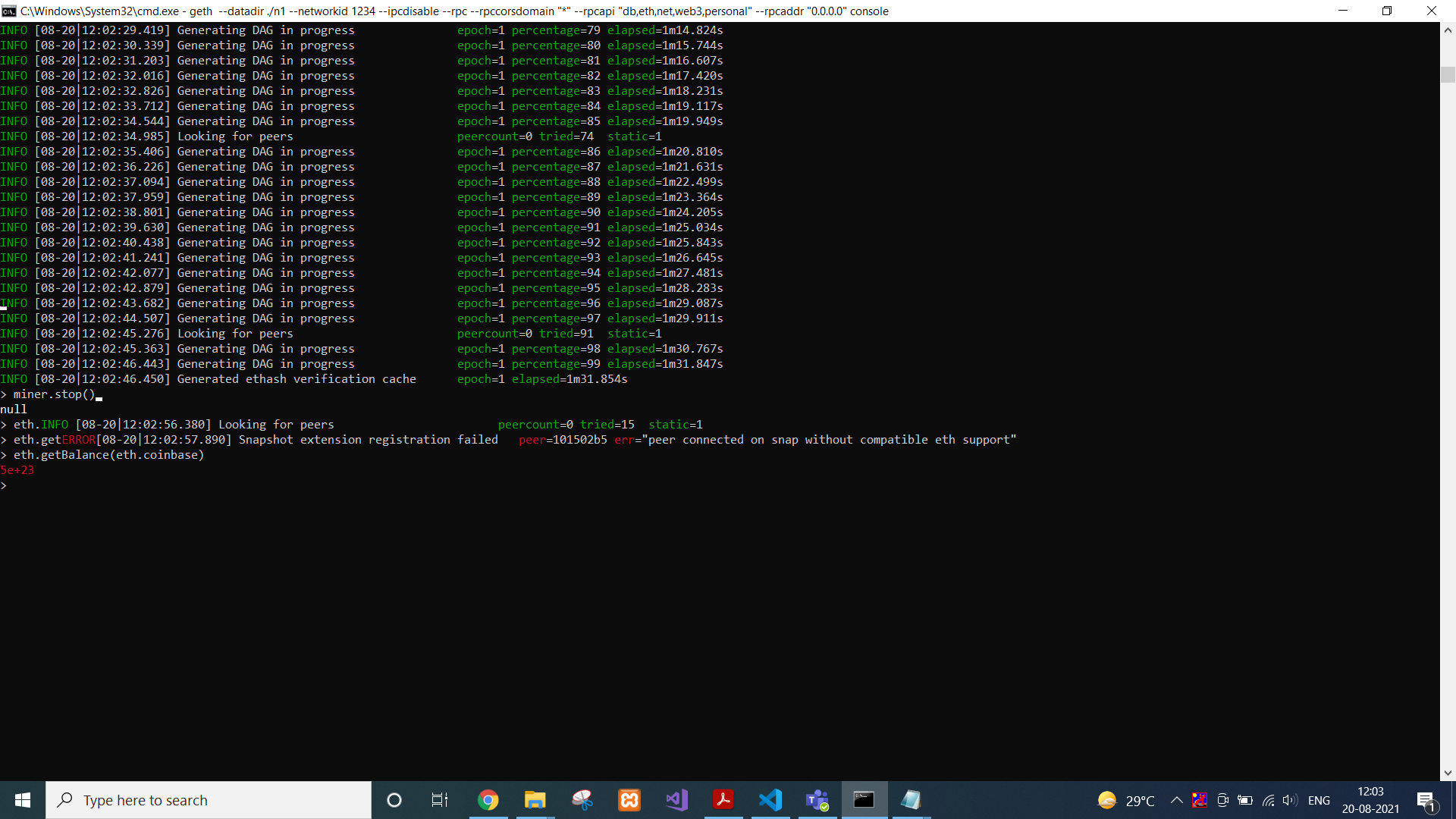
To stop that process, we can use…

miner.stop()



Now we will again check balance with following command. Ideally balance will be less than previous balance i.e. the transaction fees deducted.

eth.getBalance(eth.coinbase)



**CONCLUSION:**

In this practical, we used command prompt and geth to create our private Ethereum blockchain and we mined some Ethereum using it.