Practical 3

Blockchain Technology

2CSDE93

Dhruv Sonani

20BCE527

Date

November 08, 2022

Department of Computer Science and Engineering
Institute of Technology

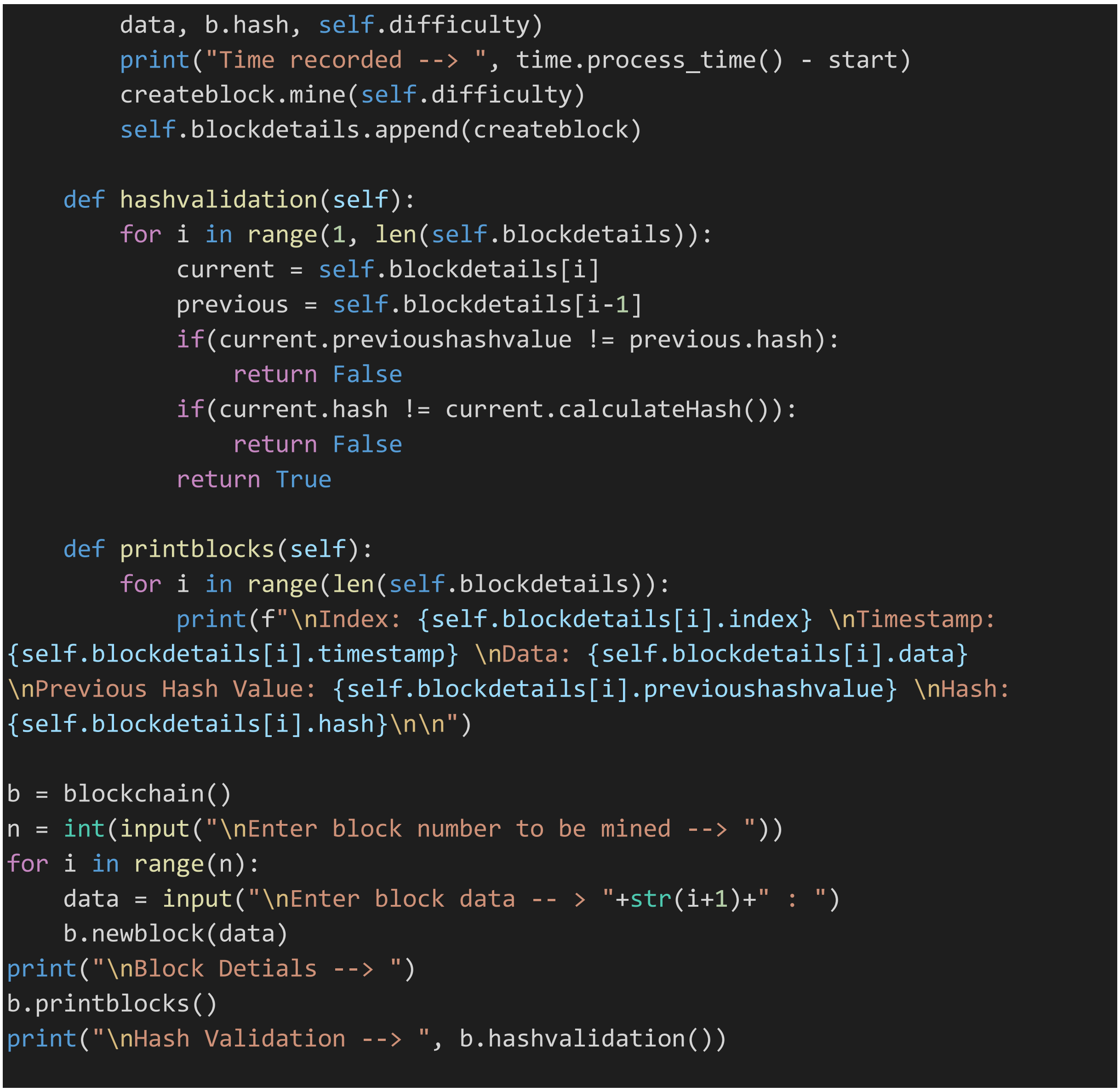
Nirma University

Ahmedabad

Aim: To perform proof of work (consensus mechanism) on previously implemented blockchain practical.

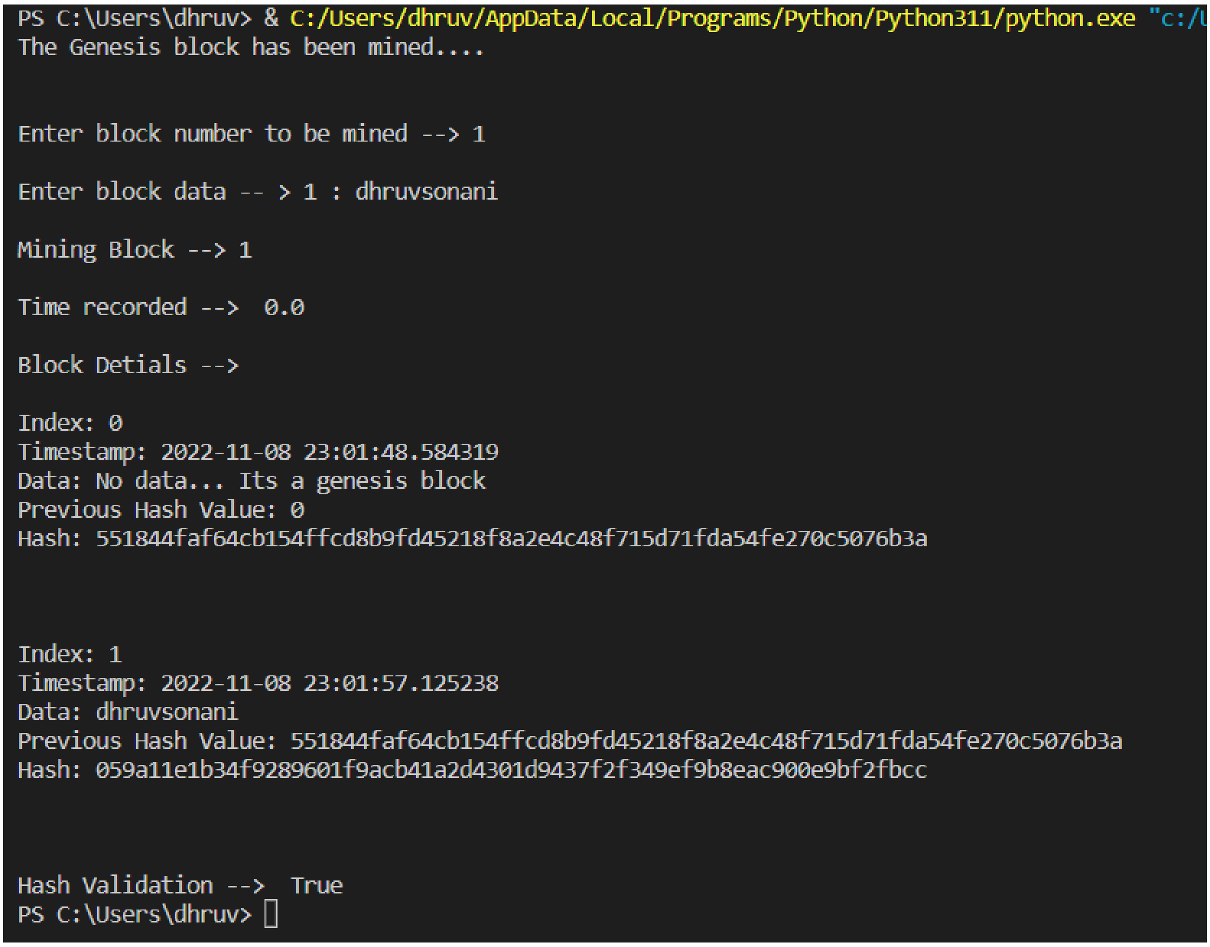
Code:

```
import hashlib
from datetime import datetime
import time

class block:
    def __init__(self, index, timestamp, data, previoushashvalue, difficulty):
        self.index = index
        self.timestamp = timestamp
        self.data = data
        self.previoushashvalue = previoushashvalue
        self.nonce = 1
        self.hash = self.calculateHash()

    def calculateHash(self):
            total = str(self.index)+str(self.timestamp)+self.data + \
            str(self.previoushashvalue)+str(self.nonce)
            return hashlib.sha256(total.encode()).hexdigest()

    def mine(self, difficulty):
            match = "0"*difficulty
            while(self.hash[0:difficulty] != match):
                self.nonce += 1
                self.hash = self.calculateHash()

class blockchain:
    def __init__(self, difficulty=1):
        self.blockdetails = []
        self.genesisblock = self.genesisblock()
        self.difficulty = difficulty

    def genesisblock(self):
        print("The Genesis block has been mined....\n")
        genesis = block(0, datetime.now(), "No data... Its a genesis block",
0, 0)
        self.blockdetails.append(genesis)

    def newblock(self, data):
        b = self.blockdetails[-1]
        start = time.process_time()
        print("\nMining Block --> "+str(b.index+1)+"\n")
        createblock = block(b.index+1, datetime.now(),
```



```
        data, b.hash, self.difficulty)
        print("Time recorded --> ", time.process_time() - start)
        createblock.mine(self.difficulty)
        self.blockdetails.append(createblock)

    def hashvalidation(self):
        for i in range(1, len(self.blockdetails)):
            current = self.blockdetails[i]
            previous = self.blockdetails[i-1]
            if(current.previoushashvalue != previous.hash):
                return False
            if(current.hash != current.calculateHash()):
                return False
            return True

    def printblocks(self):
        for i in range(len(self.blockdetails)):
            print(f"\nIndex: {self.blockdetails[i].index} \nTimestamp:
{self.blockdetails[i].timestamp} \nData: {self.blockdetails[i].data}
\nPrevious Hash Value: {self.blockdetails[i].previoushashvalue} \nHash:
{self.blockdetails[i].hash}\n\n")

b = blockchain()
n = int(input("\nEnter block number to be mined --> "))
for i in range(n):
    data = input("\nEnter block data -- > "+str(i+1)+" : ")
    b.newblock(data)
print("\nBlock Detials --> ")
b.printblocks()
print("\nHash Validation --> ", b.hashvalidation())
```

Output:



```
PS C:\Users\dhruv> & C:/Users/dhruv/AppData/Local/Programs/Python/Python311/python.exe "c:/U
The Genesis block has been mined....


Enter block number to be mined --> 1

Enter block data -- > 1 : dhruvsonani

Mining Block --> 1

Time recorded -->  0.0

Block Detials -->

Index: 0
Timestamp: 2022-11-08 23:01:48.584319
Data: No data... Its a genesis block
Previous Hash Value: 0
Hash: 551844faf64cb154ffcd8b9fd45218f8a2e4c48f715d71fda54fe270c5076b3a



Index: 1
Timestamp: 2022-11-08 23:01:57.125238
Data: dhruvsonani
Previous Hash Value: 551844faf64cb154ffcd8b9fd45218f8a2e4c48f715d71fda54fe270c5076b3a
Hash: 059a11e1b34f9289601f9acb41a2d4301d9437f2f349ef9b8eac900e9bf2fbcc



Hash Validation -->  True
PS C:\Users\dhruv> 
```