

INDEX

SR. NO		PRACTICALS	SIGN
1.		Write the following programs for Blockchain in Python:	
	A.	A simple client class that generates the private and public keys by using the built-in Python RSA algorithm and test it.	
	B.	A transaction class to send and receive money and test it.	
	C.	Create multiple transactions and display them.	
	D.	Create a blockchain, a genesis block and execute it.	
	E.	Create a mining function and test it.	
	F.	Add blocks to the miner and dump the blockchain.	
2.		Install and configure Go Ethereum and the Mist browser. Develop and test a sample application.	
3.		Implement and demonstrate the use of the following in Solidity:	
	A.	Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs, Mappings, Conversions, Ether Units, Special Variables.	
	B.	Functions, Function Modifiers, View functions, Pure Functions, Fallback Function, Function Overloading, Mathematical functions, Cryptographic functions.	
4.		Implement and demonstrate the use of the following in Solidity:	
	A.	Withdrawal Pattern, Restricted Access.	
	B.	Contracts, Inheritance, Constructors, Abstract Contracts, Interfaces.	
	C.	Libraries, Assembly, Events, Error handling.	
5.		Install hyper ledger fabric and composer. Deploy and execute the application.	
6.		Write a program to demonstrate mining of Ether.	
7.		Demonstrate the running of the blockchain node.	
8.		Demonstrate the use of Bitcoin Core API.	
9.		Create your own blockchain and demonstrate its use.	
10.		Build Apps with angular.	

Practical 1A

A) Aim: A simple client class that generates the private and public keys by using the built in Python RSA algorithm and test it.

Code:

```
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

def multiplicative_inverse(e, phi):
    for i in range(phi):
        if((e*i)%phi == 1):
            return i

def coprimes(a):
    l = []
    for x in range(2, a):
        if gcd(a, x) == 1 and multiplicative_inverse(x, phi) != None:
            l.append(x)
    for x in l:
        if x == multiplicative_inverse(x, phi):
            l.remove(x)
    return l

def generate_keypair(p, q, e, phi):
    #Use Extended Euclid's Algorithm to generate the private key
    d = multiplicative_inverse(e, phi)
    print("d=",d)
    #Public key is (e, n) and private key is (d, n)
    return ((e, n), (d, n))

def encrypt(pk, plaintext):
    #Unpack the key into its components
    key, n = pk
    #Convert each letter in the plaintext to numbers based on the character using a^b mod m
    cipher = [(ord(char) ** key) % n for char in plaintext]
    #Return the array of bytes
    return cipher

def decrypt(sk, ciphertext):
    #Unpack the key into its components
    key, n = sk
    #Generate the plaintext based on the ciphertext and key using a^b mod m
```

```

plain = [chr((char ** key) % n) for char in ciphertext]
#Return the array of bytes as a string
return ".join(plain)
if __name__ == '__main__':
    print("RSA Encrypter/ Decrypter")
    p = int(input("Enter a prime number (17, 19, 23, etc): "))
    q = int(input("Enter another prime number (Not one you entered above): "))
    n = p * q
    print("N=",n)
    phi = (p-1) * (q-1)
    print("phi_n=",phi)
    print("List of e (coprimes) :\n")
    print(str(coprimes(phi)) + "\n")
    print("Generating your public/private keypairs now . . .")
    e=int(input("Enter e: "))
    public, private = generate_keypair(p, q, e, phi)
    print("Your public key is ", public , " and your private key is ", private)
    message = str(input("Enter a message to encrypt with public key: "))
    encrypted_msg = encrypt(public, message)
    print("Your encrypted message is: ")
    print(".join(map(lambda x: str(x), encrypted_msg)))")
    print("Decrypting message with private key . . .")
    print("Your message is:")
    print(decrypt(private, encrypted_msg))

```

Output:

```

[ ] print("Decrypting message with private key . . .")
[ ] print("Your message is:")
[ ] print(decrypt(private, encrypted_msg))

RSA Encrypter/ Decrypter
Enter a prime number (17, 19, 23, etc): 19
Enter another prime number (Not one you entered above): 23
N= 437
phi_n= 396
List of e (coprimes) :

[5, 7, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49, 53, 59, 61, 65, 67, 71, 73, 79, 83, 85, 91, 95, 97, 101, 103, 107, 113, 115, 119, 125,
Generating your public/private keypairs now . . .
Enter e: 25
d= 301
Your public key is (25, 437) and your private key is (301, 437)
Enter a message to encrypt with public key: Hello
Your encrypted message is:
16585484874
Decrypting message with private key . . .
Your message is:
Hello

[ ] print((72**25)%437)
165
[ ] chr((165**301)%437)
'H'

```

Practical 1B:

Aim:

A transaction class to send and receive money and test it.

Code:

```
# following imports are required by PKI
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections

from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Cipher import PKCS1_v1_5
from collections import OrderedDict
import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5
class Client:
    def __init__(self):
        random = Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)
    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')

class Transaction:
    def __init__(self, sender, recipient, value):
        self.sender = sender
        self.recipient = recipient
        self.value = value
```

```
self.time = datetime.datetime.now()

def to_dict(self):
    if self.sender == "Genesis":
        identity = "Genesis"
    else:
        identity = self.sender.identity

    return collections.OrderedDict({
        'sender': identity,
        'recipient': self.recipient,
        'value': self.value,
        'time' : self.time})

def sign_transaction(self):
    private_key = self.sender._private_key
    signer = PKCS1_v1_5.new(private_key)
    h = SHA.new(str(self.to_dict()).encode('utf8'))
    return binascii.hexlify(signer.sign(h)).decode('ascii')

def display_transaction(transaction):
    #for transaction in transactions:
    dict = transaction.to_dict()
    print ("sender: " + dict['sender'])
    print ('----')
    print ("recipient: " + dict['recipient'])
    print ('----')
    print ("value: " + str(dict['value']))
    print ('----')
    print ("time: " + str(dict['time']))
    print ('----')
sa = Client()
rb = Client()

t1 = Transaction(
    sa,
    rb.identity,
    15.0
)
t1.sign_transaction()
display_transaction (t1)
```

Output:

```
sender: 30819f300d06092a864886f70d0101050003818d0030818902818100d18217e98615b7907512b41ed2f08e5dff9d4c2b666eee3bda991ab3f4a7af45f90eb417882e43b3ea462c6a70526f6e39ee1b3144e9017c2ccb02
-----
recipient: 30819f300d06092a864886f70d0101050003818d0030818902818100e2790fe9fe6dad786bb8a67bc0c5a34454d3765ba16f8d6699bb4bec9a2fb1833a4dac50a17b11c0eeb24cf4f7f39f5da7b32df0c46b224fc6c
-----
value: 15.0
-----
time: 2022-05-16 14:42:16.297413
-----
```

PRACTICAL 1C

Aim:

Create multiple transactions and display them.

Code:

```
# following imports are required by PKI
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Cipher import PKCS1_v1_5
from collections import OrderedDict
import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5
class Client:
    def __init__(self):
        random = Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)
    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')
class Transaction:
    def __init__(self, sender, recipient, value):
        self.sender = sender
```

```
self.recipient = recipient
self.value = value
self.time = datetime.datetime.now()

def to_dict(self):
    if self.sender == "Genesis":
        identity = "Genesis"
    else:
        identity = self.sender.identity
    return collections.OrderedDict({
        'sender': identity,
        'recipient': self.recipient,
        'value': self.value,
        'time' : self.time})

def sign_transaction(self):
    private_key = self.sender._private_key
    signer = PKCS1_v1_5.new(private_key)
    h = SHA.new(str(self.to_dict()).encode('utf8'))
    return binascii.hexlify(signer.sign(h)).decode('ascii')

def display_transaction(transaction):
    #for transaction in transactions:
    dict = transaction.to_dict()
    print ("sender: " + dict['sender'])
    print ('-----')
    print ("recipient: " + dict['recipient'])
    print ('-----')
    print ("value: " + str(dict['value']))
    print ('-----')
    print ("time: " + str(dict['time']))
    print ('-----')

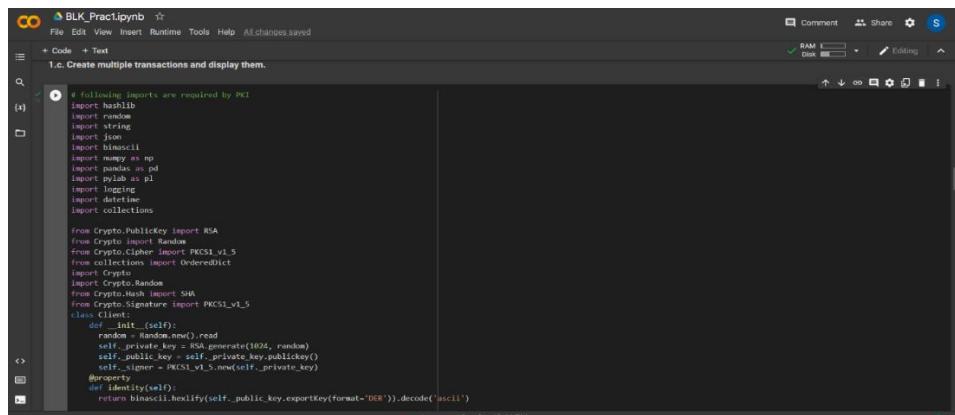
transactions = []
a = Client()
b = Client()
c = Client()
t1 = Transaction(
    a,
    b.identity,
    15.0      )
t1.sign_transaction()
transactions.append(t1)
```

```
t2 = Transaction(
    b,
    c.identity,
    25.0
)
t2.sign_transaction()
transactions.append(t2)

t3 = Transaction(
    a,
    c.identity,
    200.0
)
t3.sign_transaction()
transactions.append(t3)

tn=1
for t in transactions:#t1 t2 t3
    print("Transaction #",tn)
    display_transaction (t)
    tn=tn+1
    print ('-----')
```

Output:



The screenshot shows the Jupyter Notebook interface with two code cells. The first cell contains the Python code for creating three transactions (t1, t2, t3) and displaying them. The second cell shows the output of this code, which includes the transaction details and their corresponding signatures.

```
1.c. Create multiple transactions and display them.
[4]: # following imports are required by PKCS1_v1_5
import hashlib
import random
import string
import json
import base64
from collections import OrderedDict
from enum import IntEnum
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5
class Transaction:
    def __init__(self):
        random = Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)
    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')
```

1 completed at 12:44 PM

```
[5]: -----
[5]: Transaction # 1
sender: 30819f300d06092a854886f70b010101050003818d003081890218100b07a7a03576c296426ab79cd41b7758d38839b938bf029b722a55be39d273aa09c5de645999bead278c7e17e2617cb38e6ea5563eb27b5b11e6881e
recipient: 30819f300d06092a854886f70b010101050003818d003081890218100c84c92660ac2e11df3c1073d2a3f417b3d19feff8029e07c7c8855c5f0db428fcadbf23cf99109c1f964113aee75a76adc7ea5db29e3cf6ac49d10
value: 15.0
time: 2022-04-30 07:14:13.047265
-----
[5]: Transaction # 2
sender: 30819f300d06092a854886f70b010101050003818d003081890218100c84c92660ac2e11df3c1073d2a3f41b3d19feff8029e07c7c8855c5f0db428fcadbf23cf99109c1f964113aee75a76adc7ea5db29e3cf6ac49d10
recipient: 30819f300d06092a854886f70b010101050003818d003081890218100e8c9b099f3f3083e328e43b6d970b3ccb28435eacf9a988562f7d4dd486e40a710724a6e102819d4eb0faea4afcecd2253ab582aea44a2f1>e1b701b
value: 25.0
time: 2022-04-30 07:14:33.055079
-----
[5]: Transaction # 3
sender: 30819f300d06092a854886f70b010101050003818d003081890218100b07a7a03576c296426ab79cd41b7758d38839b722a55be39d273aa09c5de645999bead278c7e17e2617cb38e6ea5563eb27b5b11e6881e
recipient: 30819f300d06092a854886f70b010101050003818d003081890218100e8c9b09577373083e328e43b6d970b3ccb28435eacf9a988562f7d4dd486e40a710724a6e102819d4eb0faea4afcecd2253ab582aea44a2f1>e1b701b
value: 200.0
time: 2022-04-30 07:14:33.055381
-----
[5]: -----
```

1 completed at 12:44 PM

Practical 1D

Aim:

Create a block chain, a genesis block and execute it.

Code:

```
pip install pycryptodome
# following imports are required by PKI
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Cipher import PKCS1_v1_5
from collections import OrderedDict
import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5
class Client:
    def __init__(self):
        random = Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)
    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')
class Transaction:
    def __init__(self, sender, recipient, value):
        self.sender = sender
        self.recipient = recipient
        self.value = value
        self.time = datetime.datetime.now()
    def to_dict(self):
```

```

if self.sender == "Genesis":
    identity = "Genesis"
else:
    identity = self.sender.identity
return collections.OrderedDict({
    'sender': identity,
    'recipient': self.recipient,
    'value': self.value,
    'time' : self.time})
def sign_transaction(self):
    private_key = self.sender._private_key
    signer = PKCS1_v1_5.new(private_key)
    h = SHA.new(str(self.to_dict()).encode('utf8'))
    return binascii.hexlify(signer.sign(h)).decode('ascii')
def display_transaction(transaction):
    #for transaction in transactions:
    dict = transaction.to_dict()
    print ("sender: " + dict['sender'])
    print ('-----')
    print ("recipient: " + dict['recipient'])
    print ('-----')
    print ("value: " + str(dict['value']))
    print ('-----')
    print ("time: " + str(dict['time']))
    print ('-----')
def dump_blockchain (self):
    print ("Number of blocks in the chain: " + str(len (self)))
    for x in range (len(TPCoins)):
        block_temp = TPCoins[x]
        print ("block # " + str(x))
        for transaction in block_temp.verified_transactions:
            display_transaction (transaction)
            print ('-----')
            print ('=====')
class Block:
    def __init__(self):
        self.verified_transactions = []
        self.previous_block_hash = ""
        self.Nonce = ""
a = Client()
t0 = Transaction (
    "Genesis",
    a.identity,
    500.0
)

```

```
)  
  
block0 = Block()  
block0.previous_block_hash = None  
Nonce = None  
block0.verified_transactions.append (t0)  
digest = hash (block0)  
last_block_hash = digest  
TPCoins = [] #coinbase  
TPCoins.append (block0)  
dump_blockchain(TPCoins)
```

Output –

```
dump_blockchain() ->  
D Number of blocks in the chain: 1  
block # 0  
sender: Genesis  
-----  
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100daa236f845a3ec9b241ea5a3583f64f7e5c6184de1d9fba7419bb38ea65d8515d27f024021340a06f18f034003fc31c35678c56a64  
-----  
value: 500.0  
-----  
time: 2022-05-18 01:33:35.822036  
-----  
-----  
=====
```

PRACTICAL 1E:**Aim:**

Create a mining function and test it.

Code:

```
import hashlib

def sha256(message):
    return hashlib.sha256(message.encode('ascii')).hexdigest()

def mine(message, difficulty=1):
    assert difficulty >= 1
    #if(difficulty <1):
    #    return
    #'1'*3=> '111'
    prefix = '1' * difficulty
    print("prefix",prefix)
    for i in range(1000):
        digest = sha256(str(hash(message)) + str(i))
        print("testing=>" + digest)
        if digest.startswith(prefix):
            print ("after " + str(i) + " iterations found nonce: " + digest)
            return i #i= nonce value
    n=mine ("test message",3)
    print(n)
```

OUTPUT:

```
Le.py - C:\Users\Nadeem\Downloads\Le.py (3.62)
File Edit Format Run Options Window Help
import hashlib

def sha256(message):
    return hashlib.sha256(message.encode('ascii')).hexdigest()

def mine(message, difficulty=1):
    assert difficulty >= 1
    #if(difficulty <1):
    #    return
    #'1'*3=> '111'
    prefix = '1' * difficulty
    print("prefix",prefix)
    for i in range(1000):
        digest = sha256(str(hash(message)) + str(i))
        print("testing=>" + digest)
        if digest.startswith(prefix):
            print ("after " + str(i) + " iterations found nonce: " + digest)
            return i #i= nonce value
    n=mine ("test message",3)
    print(n)

Python 3.6 Shell
File Edit Shell Debug Options Window Help
testing=>jrccaa48e02adbad2301e6bc21a7401f3c8dad3e5d5f9731e866823030ua0u953e
testing=>de695bcce2e314cf19e60936014e0bac639465b126df4634c2e616c5b84e1aa
testing=>ce1d95228a946c78b75ea02c9b089a17f516243e77363dc138b8e539ddcf308
testing=>a0630b92dc346cbade8486d93a972e59ae131fa0d2505feee6a9075e981963c9
testing=>5daef4dcf4878b4ed03b5fae0fe23b4d37322bc085fa1a4bb129a6a3fc287e602cf
testing=>5daedbdbe41fcfa891377fb587123eaca75a29de6bcad168f4a376426dc368
testing=>76d3a287c3a1fe03439bf01aecc248006678f8fae4afcc6c546c3110302af52
testing=>d1f73838449ba054059356f23a69ea2125ba6bc29bd186d14fe7ed4751f56
testing=>5d389e881fd45ef276cb4b093807858ab7fae883f4680fe0aa25dd70120d1c
testing=>c0a75a0b5eec17aa350398e2768b5663e4fbcb0f13c63158ee25d63190eacb
testing=>5d389e881fd45ef276cb4b093807858ab7fae883f4680fe0aa25dd70120d1c
testing=>7dcaaf749f31ae22d7103c62be154cc9d0384e2f42b894dd0756f162358be3
testing=>8b16f230d6888e195d4d941d836dd323140bd41071583e4fa0acb64243f3692
testing=>5215ad97ee256dbfa4d704956d362d20e0e3125ct7d51d250a4cb3ce41e87d1
testing=>c6f886de970405da79afdaed357489fa0151b954bb438b8a6808778fb94dd
testing=>c1b89e981b727976a2aa56c31e39eab91696f44c2c62510259a19a7458dd845
testing=>1489e5d5d28ff68245d09b7983e26fc291b440c0e81f3b7600773babb884055
testing=>7dcaaf749f31ae22d7103c62be154cc9d0384e2f42b894dd0756f162358be3
testing=>2ea6d942bb9fd9f9cc6dbd17517d75bce04c661ea60c8590729879c8ad953c37
testing=>21a23e6cd3f7031e7566c40f2e5824642ae71f6cf04d931ff15bba786d835bf
testing=>df4d1e2d1605902dbc732629000261429157be6559dc5140ff494a827f33
testing=>d23af9c7618217c850b615302833d2d207f347105cf9611e97df3479b288c4
testing=>ed23af9c7618217c850b615302833d2d207f347105cf9611e97df3479b288c4
testing=>984d6398e9558c8d66878108d55de393tbaa8f2ffdfa1be637c8a34cf0d1a59
testing=>fc15305b04f9469c2f48b9a7fd156299e3d1f8c7b51034ab7d42587fe05d64
testing=>ce5e5673358293cfd1f450f39947c90abbe8f95844d6afdb5974b2d19378a0
testing=>1389f298a09e9438d90d708261dbc0966767dafb1fdaf7b7e01b1ed548fee08
testing=>4171ledc90562df8d12e8a446c174aab28fa355d9fa0dda690e5b15ecc7f0f4
testing=>d3ab47766e7f35392c1e18686a84659f375bd28128fb09ec2b9f6cccd10ef90f
testing=>f20ca7e407a3563a0a7207655ff7cc3d85062e73f480d9eefaaef04b9e47a
testing=>9d9e0b5f50300eee615d300e64bc09b27a8cac007df5c5f29ff8942afc9e6
testing=>8a59d619dcda461c6190a9553eaf846804263addf709947c065a0c8d47f03e
testing=>edb865a9d031372df5b13233d8e3e9f59464b52df216610e2a972b54a3e3d12
testing=>ac5ce984c22c901e0de995ca3214a4d49d5f4c29a635ccc909bf8c2e81090
testing=>a9a5e3125edacc455f194709c6c0cb2d4675693f7b204f40803b48832a9d9b37
testing=>4c1491b7d5e919d394e8379102ff4d7e49a5a9015996ac80e142cabd326c2422
None
>>>
```

Go to Settings to activate Windows.
Lm: 879 Col: 73
32°C Smoke 15:58 PM 5/14/2022

PRACTICAL 1F:

Aim:

Add blocks to the miner and dump the blockchain.

Code:

```
# following imports are required by PKI
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections

!pip install pycryptodome
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Cipher import PKCS1_v1_5
from collections import OrderedDict
import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5

class Client:
    def __init__(self):
        random = Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)
    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')
```

```
class Transaction:  
    def __init__(self, sender, recipient, value):  
        self.sender = sender  
        self.recipient = recipient  
        self.value = value  
        self.time = datetime.datetime.now()  
  
    def to_dict(self):  
        if self.sender == "Genesis":  
            identity = "Genesis"  
        else:  
            identity = self.sender.identity  
  
        return collections.OrderedDict({  
            'sender': identity,  
            'recipient': self.recipient,  
            'value': self.value,  
            'time' : self.time})  
  
    def sign_transaction(self):  
        private_key = self.sender._private_key  
        signer = PKCS1_v1_5.new(private_key)  
        h = SHA.new(str(self.to_dict()).encode('utf8'))  
        return binascii.hexlify(signer.sign(h)).decode('ascii')  
  
def display_transaction(transaction):  
    #for transaction in transactions:  
    dict = transaction.to_dict()  
    print ("sender: " + dict['sender'])  
    print ('-----')  
    print ("recipient: " + dict['recipient'])  
    print ('-----')  
    print ("value: " + str(dict['value']))  
    print ('-----')  
    print ("time: " + str(dict['time']))  
    print ('-----')  
  
def dump_blockchain (self):
```

```
print ("Number of blocks in the chain: " + str(len (self)))
for x in range (len(TPCoins)):
    block_temp = TPCoins[x]
    print ("block # " + str(x))
    for transaction in block_temp.verified_transactions:
        display_transaction (transaction)
        print ('-----')
    print ('=====')\n\n

class Block:
    def __init__(self):
        self.verified_transactions = []
        self.previous_block_hash = ""
        self.Nonce = ""\n\n

def sha256(message):
    return hashlib.sha256(message.encode('ascii')).hexdigest()\n\n

def mine(message, difficulty=1):
    assert difficulty >= 1
    #if(difficulty <1):
    #    return
    #'1'*3=> '111'
    prefix = '1' * difficulty
    for i in range(1000):
        digest = sha256(str(hash(message)) + str(i))
        if digest.startswith(prefix):
            return i #i= nonce value\n\n

Ankit = Client()
Yash =Client()
Aayush =Client()
t0 = Transaction (
    "Genesis",
    Ankit.identity,
    500.0
)
t1 = Transaction (
```

```
Yash,  
Ankit.identity,  
40.0  
)  
t2 = Transaction (  
    Yash,  
    Ankit.identity,  
    70.0  
)  
t3 = Transaction (  
    Aayush,  
    Yash.identity,  
    700.0  
)  
#blockchain  
TPCoins = []  
  
block0 = Block()  
block0.previous_block_hash = None  
Nonce = None  
block0.verified_transactions.append (t0)  
digest = hash (block0)  
last_block_hash = digest #last_block_hash it is hash of block0  
TPCoins.append (block0)  
  
block1 = Block()  
block1.previous_block_hash = last_block_hash  
block1.verified_transactions.append (t1)  
block1.verified_transactions.append (t2)  
block1.Nonce=mine (block1, 2)  
digest = hash (block1)  
last_block_hash = digest  
TPCoins.append (block1)  
  
block2 = Block()  
block2.previous_block_hash = last_block_hash  
block2.verified_transactions.append (t3)  
Nonce = mine (block2, 2)  
block2.Nonce=mine (block2, 2)
```

```

digest = hash(block2)
last_block_hash = digest
TPCoins.append(block2)

dump_blockchain(TPCoins)

```

OUTPUT:

```

Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
RAM Disk Editing
+ Code + Text
Number of blocks in the chain: 3
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d0101050003818d0030818902818100d64f1af07c88850aed1d5041cc6cc956766e0173c3ec13d3f59d6b81e4dd182cd7b40e77ed48dc2fc2e3eb78774313a10e89b5889ffdd6b54ff3268ac788130f5a2e895c95629bead4cd2989f3eb22365
value: 500.0
time: 2022-05-17 18:44:31.810453
-----
block # 1
sender: 30819f300d06092a864886f70d0101050003818d0030818902818100d64f1af07c88850aed1d5041cc6cc956766e0173c3ec13d3f59d6b81e4dd182cd7b40e77ed48dc2fc2e3eb78774313a10e89b5889ffdd6b54ff3268ac788130f5a2e895c95629bead4cd2989f3eb22365
recipient: 30819f300d06092a864886f70d0101050003818d0030818902818100d64f1af07c88850aed1d5041cc6cc956766e0173c3ec13d3f59d6b81e4dd182cd7b40e77ed48dc2fc2e3eb78774313a10e89b5889ffdd6b54ff3268ac788130f5a2e895c95629bead4cd2989f3eb22365
value: 40.0
time: 2022-05-17 18:44:31.810750
-----
block # 2
sender: 30819f300d06092a864886f70d0101050003818d0030818902818100d64f1af07c88850aed1d5041cc6cc956766e0173c3ec13d3f59d6b81e4dd182cd7b40e77ed48dc2fc2e3eb78774313a10e89b5889ffdd6b54ff3268ac788130f5a2e895c95629bead4cd2989f3eb22365
recipient: 30819f300d06092a864886f70d0101050003818d0030818902818100d64f1af07c88850aed1d5041cc6cc956766e0173c3ec13d3f59d6b81e4dd182cd7b40e77ed48dc2fc2e3eb78774313a10e89b5889ffdd6b54ff3268ac788130f5a2e895c95629bead4cd2989f3eb22365
value: 70.0
time: 2022-05-17 18:44:31.811029
-----

```

3s completed at 12:14 AM

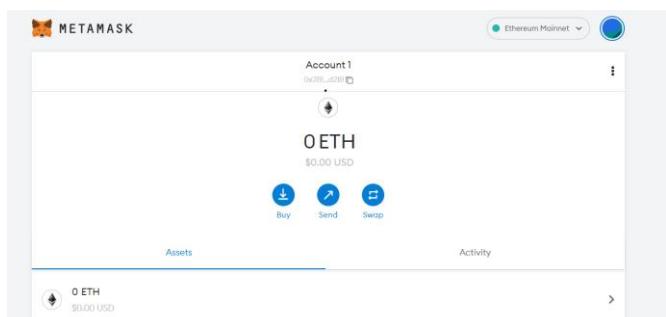
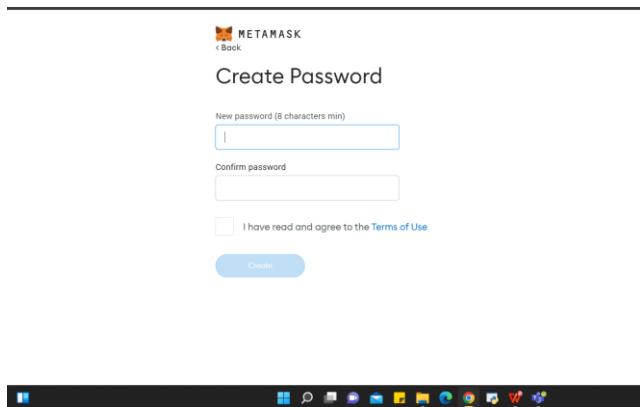
Practical 2

Aim:

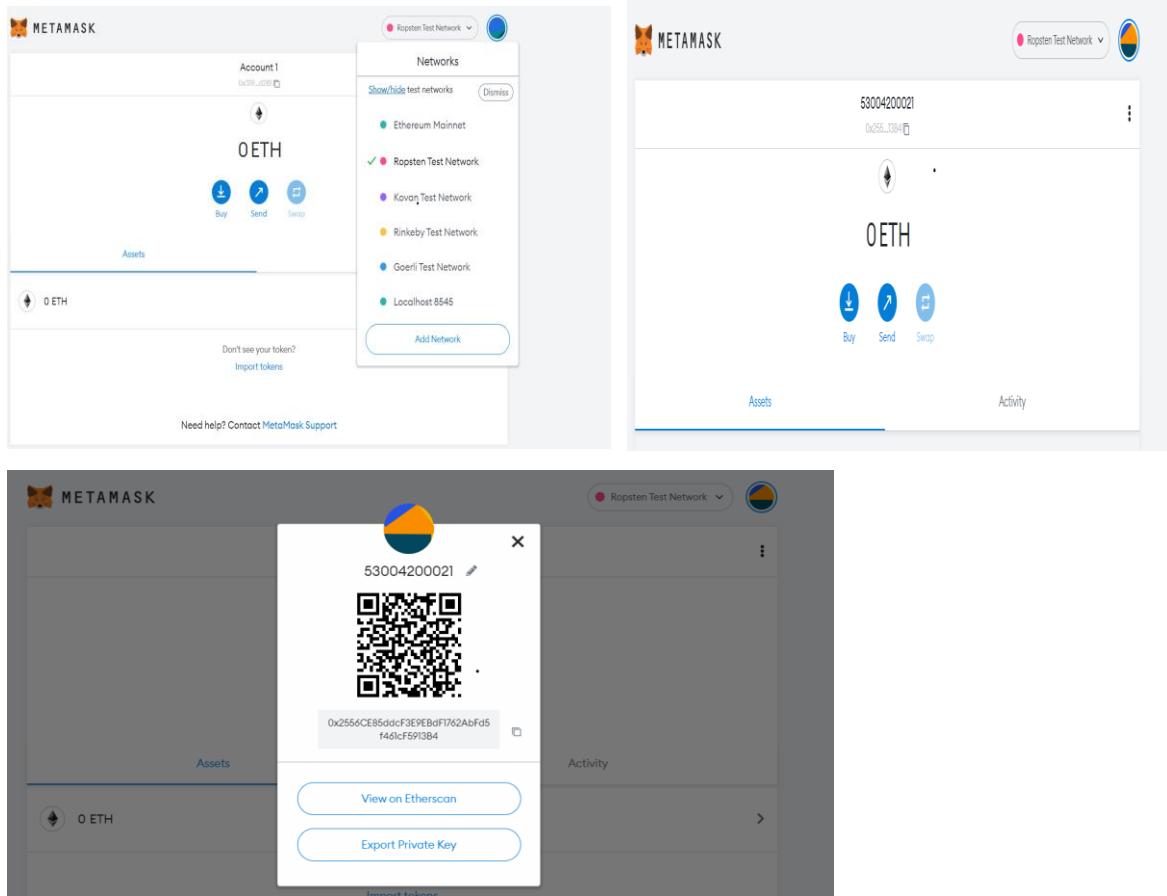
Install and configure Go Ethereum and the Mist browser. Develop and test a sample application

A) Ethereum transaction with the help of smart contract in Remix IDE with Ropsten test net.
Note:Mist was deprecated in March 2019 after developers decided other browser developers and wallet makers were better able to create products for this quickly evolving space.So in this Remix IDE is used instead of Mist Browser

- 1) Create a account in Meta Mask.



- 2) To deploy the contract, we need an account and with some ether on the Ropsten test network. To open Ropsten test network you first need to enable the option to show test networks



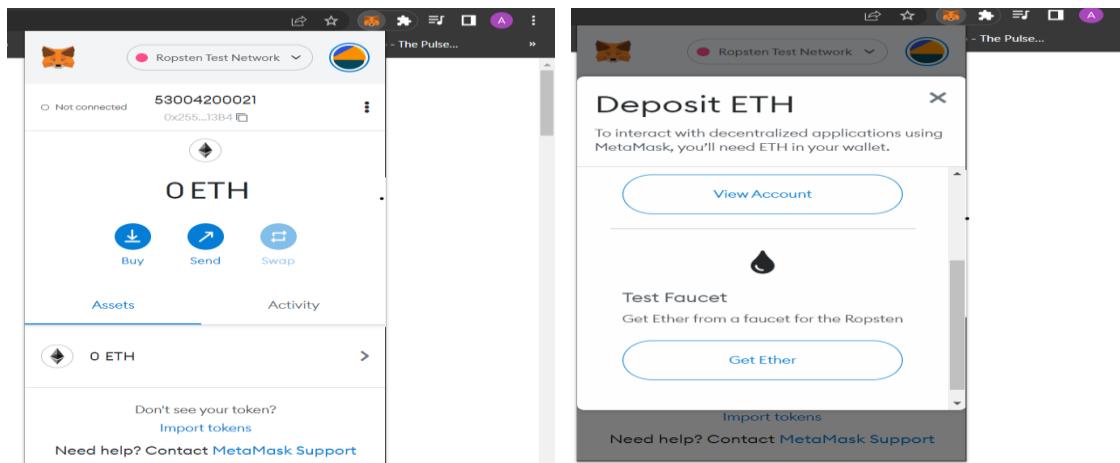
3) Request Free ether from [here](#) OR [here](#). Paste the copied account address in text box and click on send me 1 test ether.

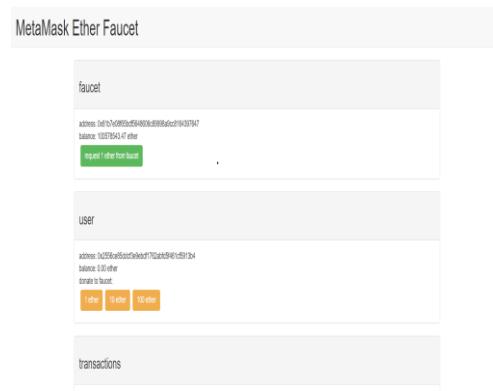
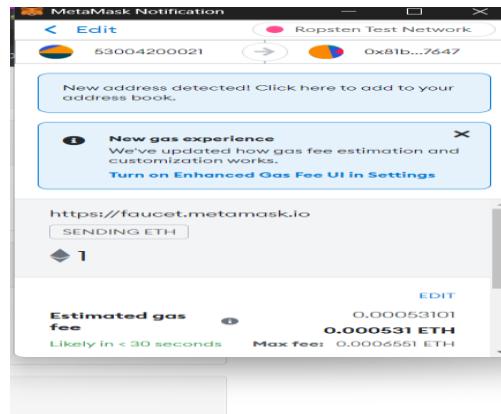
Click the MetaMask icon near the top right corner of the browser window.

Select the "Ropsten Test Net" in the dropdown list on the top.

Click "BUY" and "ROPSTEN TEST FAUCET". MetaMask will bring you the <https://faucet.metamask.io> site with your account address displayed.

Click "Request 1 ether from faucet". You will see the transfer transaction ID displayed at the bottom.



Ropsten Ethereum (rETH) Faucet

Receive 1 rETH per request

Enter Your Ropsten Address
0x2556CE85ddcF3E9EBdF1762AbFd5f461cF5913B4

I am human hCaptcha [Privacy - Terms](#)

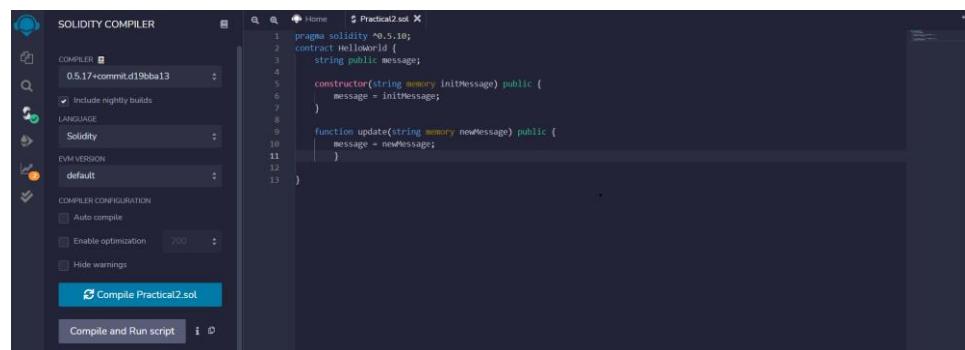
Send Ropsten ETH

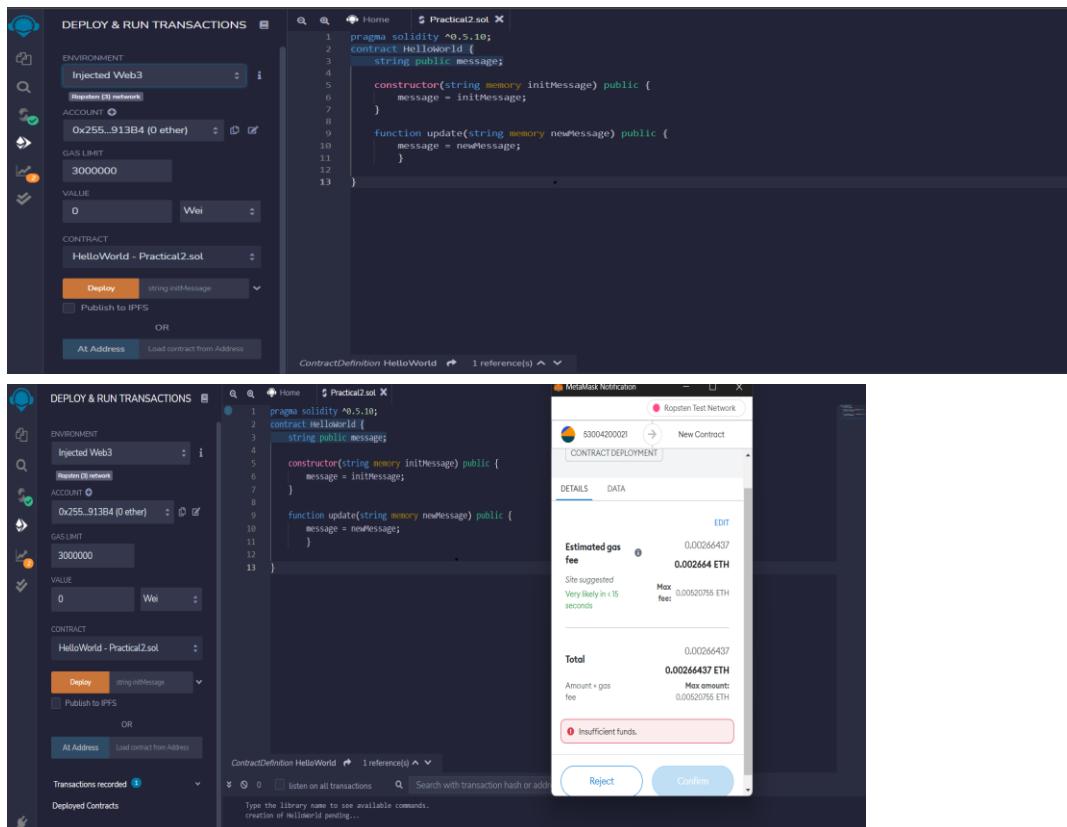
5253412 ETH left in Faucet, Gas Limit 400K

- 4) Write the code in remix IDE, by creating a new file we have connected meta mask to the ropsten testnet with an account that has some ether.

Code :

```
pragma solidity ^0.5.10;
contract HelloWorld {
    string public message;
    constructor(string memory initMessage) public {
        message = initMessage;
    }
    function update(string memory newMessage) public {
        message = newMessage;
    }
}
```

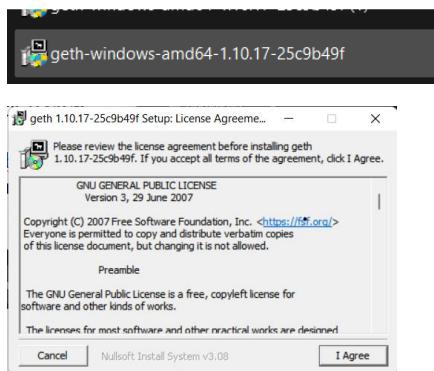




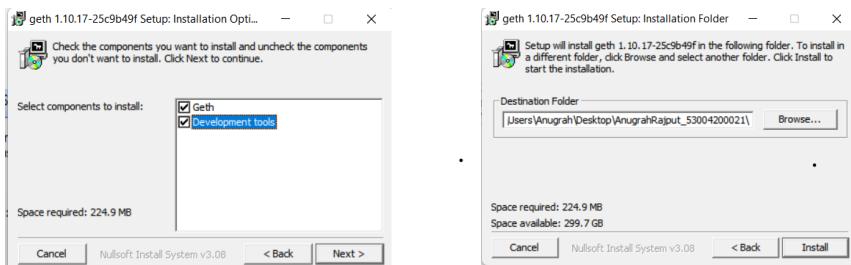
5) Click on the contract deployment — it should open up the etherscan page to look at our transaction details.

B) Install and Configure Geth and creating a private blockchain and connecting with Remix

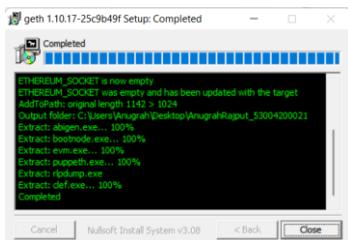
1) Download geth node from <https://geth.ethereum.org/downloads/> double click on the .exe file. Click on Agree.



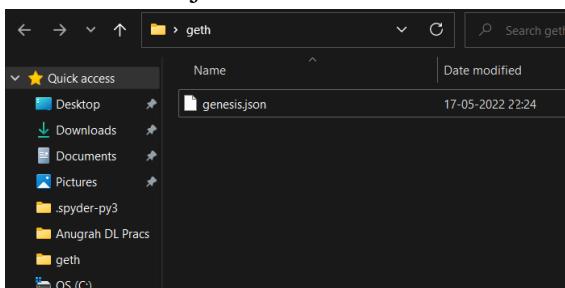
- 2) Click on I Agree
- 3) Click on Next
- 4) Specify the path and click on install.



5) Once installed open command prompt write geth version to see if installation is done properly



6) Create a geth folder in that Open a notepad and write the code to create a genesis block and save it as .json file

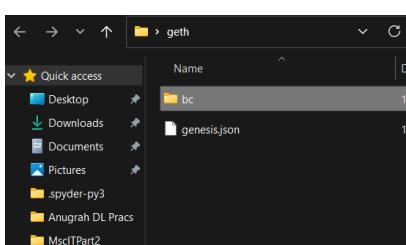


7) Run this command to create the block chain :

```
geth --datadir C:\Users\Anugrah\Desktop\geth\bc init
```

```
C:\Users\Anugrah\Desktop\geth\genesis.json
```

Now we will find a blockchain file in the geth folder.



8) Now run this command to activate the blockchain

```
geth --identity "localB" --rpc --rpcport "8280" --rpccorsdomain "*" -- rpcapi "db,eth,net,web3" --datadir "C:\Users\Anugrah\Desktop\geth\bc" - -port "30303" -- nodiscover --networkid 1999 console
```

9) Create a account and check the balance of the using the following code :

```
personal.newAccount()
```

```
eth.getBalance("the account number generated while creating the account")
```

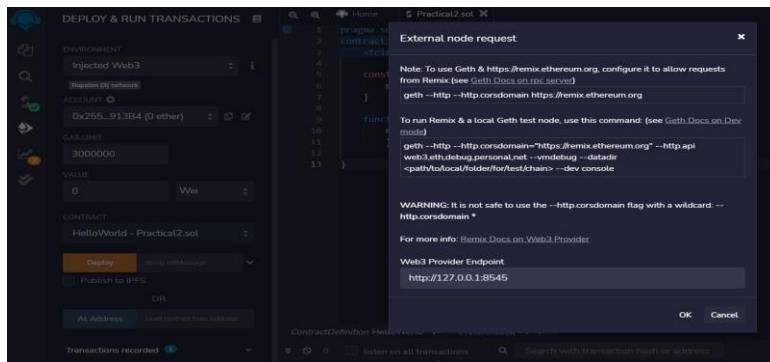
10) Now start the mining using **miner.start()** command

Now we can connect this blockchain to remix IDE with the help of the port that we have

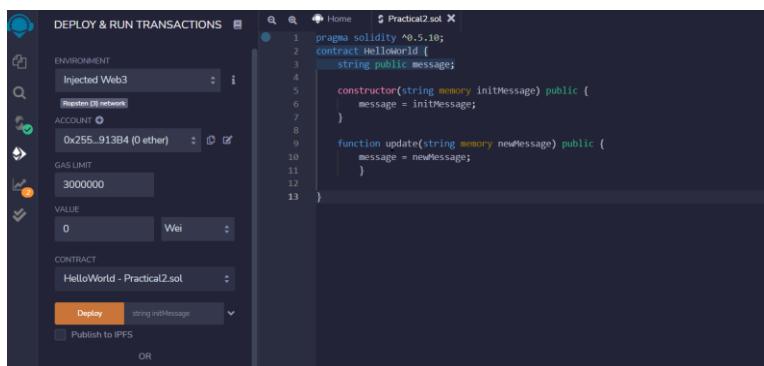
specified in the genesis file

11) Click on deploy and run transactions in Remix IDE click on Injected Web 3

Specify the port number



Now you can see the account number has changed and now you can easily deploy the smart contracts and do transactions.



PRACTICAL 3A

AIM:

WRITE A SOLIDITY PROGRAM FOR VARIABLES, OPERATORS, LOOPS, DECISION MAKING AND STRING.

A)Variables:

supports three types of variables.

State Variables – Variables whose values are permanently stored in a contract storage.

Local Variables – Variables whose values are present till function is executing.

Global Variables – Special variables exists in the global namespace used to get information about the blockchain.i.e. blockhash(uint blockNumber) returns (bytes32), block.coinbase (address payable), block.difficulty (uint).....and many more

Code:

```
pragma solidity ^0.5.0;

contract SolidityTest {

    uint storedData; // State variable

    constructor() public {
        storedData = 10;
    }

    function getResult() public view returns(uint){
        uint a = 1; // local variable
        uint b = 2;
        uint result = a + b;
        return result; //access the state variable}
    }
}
```

The screenshot shows the Remix Ethereum IDE interface. On the left, there's a sidebar titled "SOLIDITY STATIC ANALYSIS" with sections like Security, Gas & Economy, ERC, and Miscellaneous. The main area displays the following Solidity code:

```

pragma solidity ^0.5.0;
contract SolidityTest {
    uint storedData; // State variable
    constructor() public {
        storedData = 10;
    }
    function getStored() public view returns (uint) {
        uint a = 11; // Local variable
        uint result = a + b;
        return result; // Access the state variable
    }
}

```

Below the code, the terminal window shows a transaction log:

```

+ remix
Type the library name to see available commands.
Creation of SolidityTest pending...
[tx] from: 0x0...3edc4 to: SolidityTest.(constructor) value: 0 wei data: 0x00...10032 logs: 0 hash: 0x00...a7c05
call to SolidityTest.getStored()
[tx] from: 0x00000000000000000000000000000000 to: SolidityTest.getStored() data: 0x002...927e9

```

1.State Variable:

```

// Solidity program to demonstrate state variables

// pragma solidity ^0.5.0;

// Creating a contract

contract Solidity_var_Test {

    // Declaring a state variable

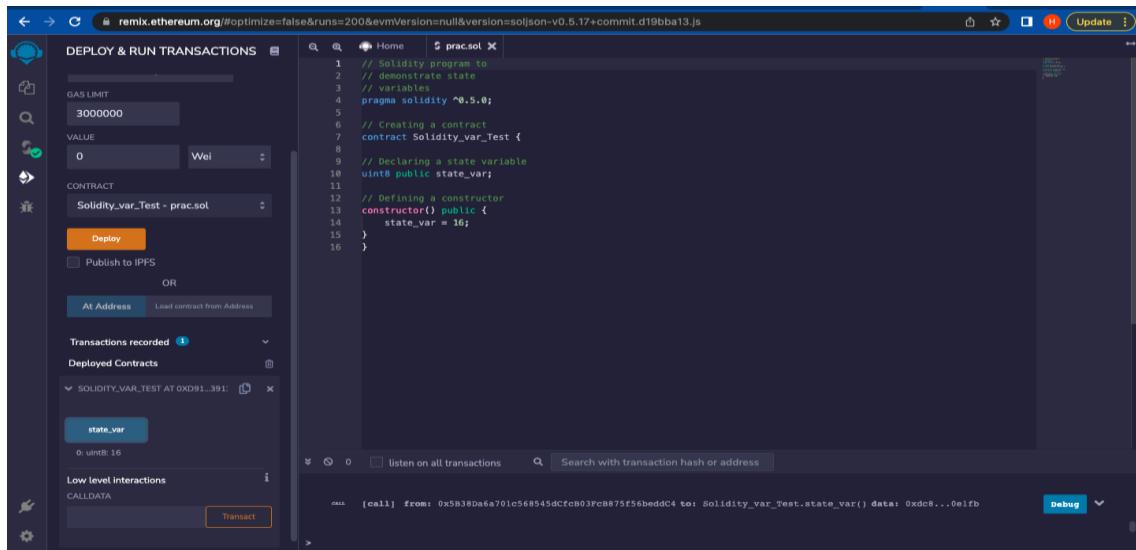
    uint8 public state_var;

    // Defining a constructor

    constructor() public {

        state_var = 16;
    }
}

```



2.Local Variable:

```
// Solidity program to demonstrate
// local variables
pragma solidity ^0.5.0;

// Creating a contract
contract Solidity_var_Test {

// Defining function to show the declaration and
// scope of local variables
function getResult() public view returns(uint){

// Initializing local variables
uint local_var1 = 1;
uint local_var2 = 2;
uint result = local_var1 + local_var2;

// Access the local variable
return result;
}
}
```

```

// Solidity program to
// show Global variables

pragma solidity ^0.5.0;

// Creating a contract

contract Solidity_var_Test {
    // Defining function to show the declaration and
    // scope of local variables
    function getResult() public view returns(uint) {
        // Initializing local variables
        uint local_var1 = 1;
        uint local_var2 = 2;
        uint result = local_var1 + local_var2;
        // Access the local variable
        return result;
    }
}

```

3.Global variable:

```

// Solidity program to
// show Global variables

pragma solidity ^0.5.0;

// Creating a contract

contract Test {
    // Defining a variable
    address public admin;

    // Creating a constructor to
    // use Global variable

    constructor() public {
        admin = msg.sender;
    }
}

```

```

// Solidity program to
// show Global variables
pragma solidity ^0.5.0;

contract Test {
    address public admin;
    constructor() public {
        admin = msg.sender;
    }
}

```

Scope of local variables is limited to function in which they are defined but State variables can have three types of scopes.

Public – Public state variables can be accessed internally as well as via messages. For a public state variable, an automatic getter function is generated.

Internal – Internal state variables can be accessed only internally from the current contract or contract deriving from it without using this.

Private – Private state variables can be accessed only internally from the current contract they are defined not in the derived contract from it.

B) Operators

Solidity supports the following types of operators.

Arithmetic Operators

Comparison Operators

Logical (or Relational) Operators

Assignment Operators

Conditional (or ternary) Operators

1. Arithematic Operator

```
// Solidity contract to demonstrate  
  
// Arithematic Operator  
pragma solidity ^0.5.0;  
  
// Creating a contract  
contract SolidityTest {  
  
    // Initializing variables  
  
    uint16 public a = 20;  
  
    uint16 public b = 10;  
  
    // Initializing a variable  
  
    // with sum  
  
    uint public sum = a + b;  
  
    // Initializing a variable  
  
    // with the difference  
  
    uint public diff = a - b;  
  
    // Initializing a variable  
  
    // with product  
  
    uint public mul = a * b;  
  
    // Initializing a variable  
  
    // with quotient  
  
    uint public div = a / b;  
  
    // Initializing a variable  
  
    // with modulus  
  
    uint public mod = a % b;  
  
    // Initializing a variable  
  
    // decrement value  
  
    uint public dec = --b;  
  
    // Initializing a variable  
  
    // with increment value
```

```
uint public inc = ++a;
}
```

OUTPUT:

```
// Solidity contract to demonstrate
// Arithmetic Operator
pragma solidity ^0.5.0;

// Creating a contract
contract SolidityTest {
    // Initializing variables
    uint16 public a = 20;
    uint16 public b = 10;

    // Initializing a variable
    // with sum
    uint public sum = a + b;

    // Initializing a variable
    // with the difference
    uint public diff = a - b;

    // Initializing a variable
    // with product
    uint public mul = a * b;

    // Initializing a variable
    // with quotient
    uint public div = a / b;

    // Initializing a variable
    // with modulus
    uint public mod = a % b;
}

// Initializing a variable
// decrement value

```

2.Relational Operator

```
// Solidity program to demonstrate

// Relational Operator

pragma solidity ^0.5.0;

// Creating a contract

contract SolidityTest {

    // Declaring variables

    uint16 public a = 20;

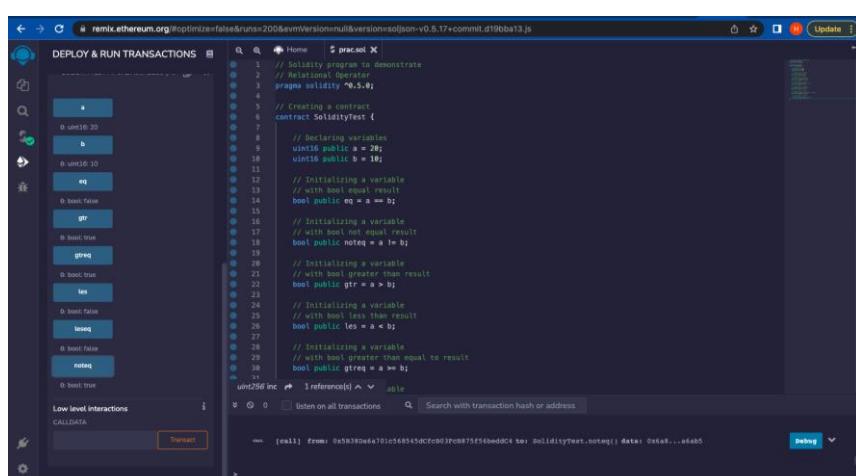
    uint16 public b = 10;

    // Initializing a variable

    // with bool equal result
```

```
bool public eq = a == b;  
// Initializing a variable  
// with bool not equal result  
bool public noteq = a != b;  
// Initializing a variable  
// with bool greater than result  
bool public gtr = a > b;  
  
// Initializing a variable  
// with bool less than result  
bool public les = a < b;  
  
// Initializing a variable  
// with bool greater than equal to result  
bool public gtreq = a >= b;  
  
// Initializing a variable  
// bool less than equal to result  
bool public leseq = a <= b;  
}
```

OUTPUT:



3.Logical Operators

```
// Solidity program to demonstrate
// Logical Operators
pragma solidity ^0.5.0;

// Creating a contract
contract logicalOperator{
    // Defining function to demonstrate
    // Logical operator
    function Logic(
        bool a, bool b) public view returns(
        bool, bool, bool){

        // Logical AND operator
        bool and = a&&b;

        // Logical OR operator
        bool or = a||b;

        // Logical NOT operator
        bool not = !a;

        return (and, or, not);}}}
```

OUTPUT:

The screenshot shows the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar displays the deployed contract 'logicalOperator - pracsol'. It has options to 'Deploy', 'Publish to IPFS', and 'At Address'. Below it, 'Transactions recorded' and 'Deployed Contracts' sections are visible. A dropdown menu for the contract 'Logic' shows the three functions: 'and', 'or', and 'not'. The main workspace shows the Solidity code for the contract. At the bottom, the 'Low level interactions' section shows a transaction being called from address 0x50340e6a701c568545dCfca03re875f54bed0C4 to the contract's address, with the function 'logicalOperator.Logic(bool,bool)' and a value of 0x00000000.

4.Bitwise Operators

```
// Solidity program to demonstrate  
// Bitwise Operator  
pragma solidity ^0.5.0;
```

```
// Creating a contract
```

```
contract SolidityTest {
```

```
// Declaring variables
```

```
uint16 public a = 20;
```

```
uint16 public b = 10;
```

```
// Initializing a variable
```

```
// to '&' value
```

```
uint16 public and = a & b;
```

```
// Initializing a variable
```

```
// to '|' value
```

```
uint16 public or = a | b;
```

```
// Initializing a variable
```

```
// to '^' value
```

```
uint16 public xor = a ^ b;
```

```
// Initializing a variable
```

```
// to '<<' value
```

```
uint16 public leftshift = a << b;
```

```
// Initializing a variable
```

```
// to '>>' value
uint16 public rightshift = a >> b;

// Initializing a variable
// to '~' value
uint16 public not = ~a ;
}
```

OUTPUT:

The screenshot shows the Remix Ethereum IDE interface. On the left, there's a sidebar with various operations like 'and', 'not', 'or', etc., each with a corresponding button. The main area displays the Solidity code for a contract named 'prac.sol'. The code includes comments explaining the purpose of each line. Below the code, the assembly output is shown in a large text area. At the bottom, there's a transaction log and a 'debug' button.

```
9  uint16 public a = 20;
10 uint16 public b = 10;
11 // Initializing a variable
12 // to '&' value
13 uint16 public and = a & b;
14
15 // Initializing a variable
16 // to '!' value
17 uint16 public or = a | b;
18
19 // Initializing a variable
20 // to '^' value
21 uint16 public xor = a ^ b;
22
23 // Initializing a variable
24 // to '<<' value
25 uint16 public leftshift = a << b;
26
27 // Initializing a variable
28 // to '>>' value
29 uint16 public rightshift = a >> b;
30
31 // Initializing a variable
32 // to '~' value
33 uint16 public not = ~a ;
34
35
36
37 }
```

5. Assignment Operator

```
// Solidity program to demonstrate
```

```
// Assignment Operator
```

```
pragma solidity ^0.5.0;
```

```
// Creating a contract
```

```
contract SolidityTest {
```

```
// Declaring variables
```

```
uint16 public assignment = 20;
```

```
uint public assignment_add = 50;
```

```
uint public assign_sub = 50;
```

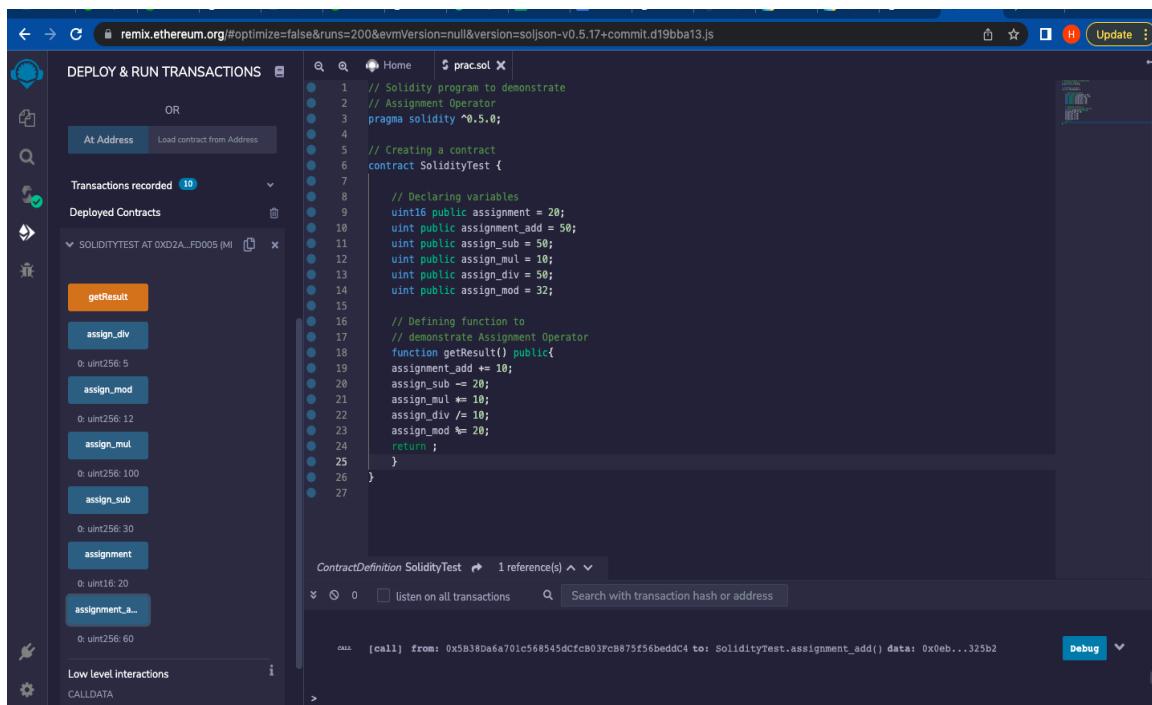
```

uint public assign_mul = 10;
uint public assign_div = 50;
uint public assign_mod = 32;

// Defining function to
// demonstrate Assignment Operator
function getResult() public{
    assignment_add += 10;
    assign_sub -= 20;
    assign_mul *= 10;
    assign_div /= 10;
    assign_mod %= 20;
    return ;
}
}

```

OUTPUT:



The screenshot shows the Remix Ethereum IDE interface. On the left, there's a sidebar with icons for deploying contracts, loading contracts, and interacting with them. The main area has tabs for "pracsol" and "Home". The "pracsol" tab is active, displaying the Solidity code. The code defines a contract named SolidityTest with variables assignment, assignment_add, assignment_sub, assignment_mul, assignment_div, and assignment_mod. It includes a function getResult() that performs various arithmetic operations on these variables. To the right of the code, there's a preview pane showing the state of the variables after the function execution. Below the code, the "ContractDefinition SolidityTest" section shows a transaction log with a call to assignment_add(). At the bottom, there are buttons for "debug" and "Low level interactions".

```

1 // Solidity program to demonstrate
2 // Assignment Operator
3 pragma solidity ^0.5.0;
4
5 // Creating a contract
6 contract SolidityTest {
7
8     // Declaring variables
9     uint16 public assignment = 20;
10    uint public assignment_add = 50;
11    uint public assignment_sub = 10;
12    uint public assignment_mul = 10;
13    uint public assignment_div = 50;
14    uint public assignment_mod = 32;
15
16    // Defining function to
17    // demonstrate Assignment Operator
18    function getResult() public{
19        assignment_add += 10;
20        assign_sub -= 20;
21        assign_mul *= 10;
22        assign_div /= 10;
23        assign_mod %= 20;
24        return ;
25    }
26
27 }

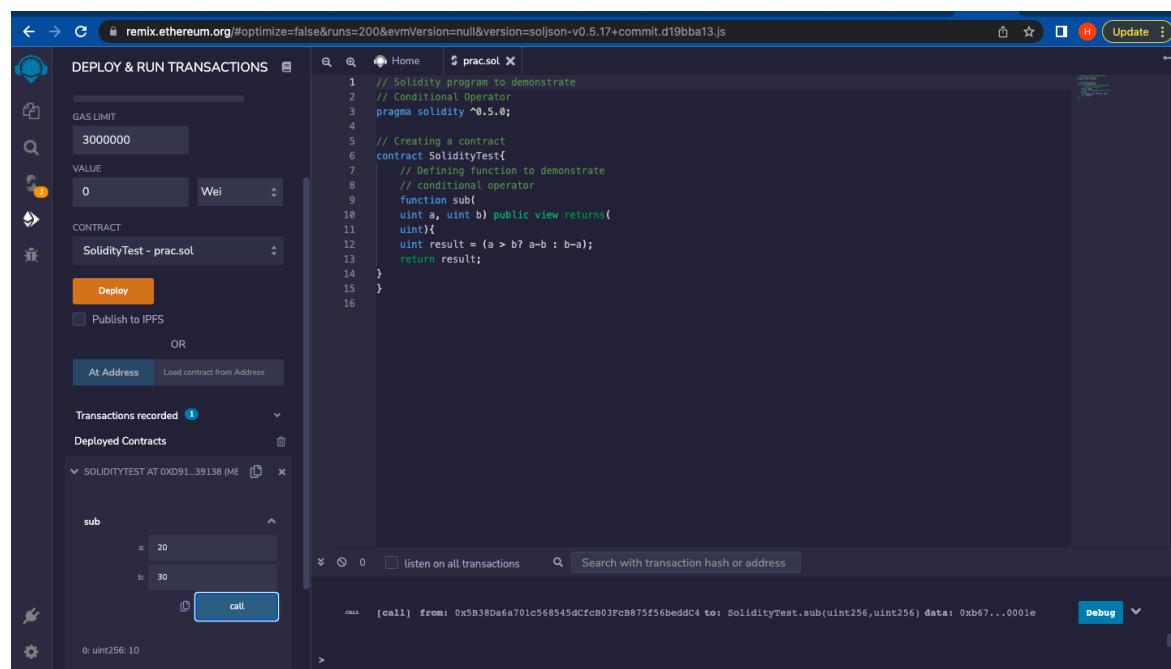
ContractDefinition SolidityTest 1 reference(s) ▾
CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: SolidityTest.assignment_add() data: 0x0eb...325b2

```

6. Conditional Operators

```
// Solidity program to demonstrate  
// Conditional Operator  
pragma solidity ^0.5.0;  
  
// Creating a contract  
contract SolidityTest{  
    // Defining function to demonstrate  
    // conditional operator  
    function sub(  
        uint a, uint b) public view returns(  
        uint){  
        uint result = (a > b? a-b : b-a);  
        return result;  
    }  
}
```

OUTPUT:



The screenshot shows the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is open, displaying deployment settings: GAS LIMIT set to 3000000, VALUE set to 0 Wei, and CONTRACT set to SolidityTest - prac.sol. Below these are buttons for 'Deploy' and 'Publish to IPFS'. In the center, the code editor shows the Solidity code provided in the text block. At the bottom, the transaction history section shows a single transaction: a call from address 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to the deployed contract at address 0xD91...39138, with a value of 0 and a data payload of 0xb67...0001e. A 'call' button is visible next to the transaction details.

C)Loops:

1.While loop: The most basic loop in Solidity is the **while** loop which would be discussed in this chapter. The purpose of a **while** loop is to execute a statement or code block repeatedly as long as an **expression** is true. Once the expression becomes **false**, the loop terminates.

2.do-while loop: The **do...while** loop is similar to the **while** loop except that the condition check happens at the end of the loop. This means that the loop will always be executed at least once, even if the condition is **false**.

3.for loop: The **for** loop is the most compact form of looping. It includes the following three important parts –

The **loop initialization** where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.

The **test statement** which will test if a given condition is true or not. If the condition is true, then the code given inside the loop will be executed, otherwise the control will come out of the loop.

The **iteration statement** where you can increase or decrease your counter.

4.loop control: Solidity provides full control to handle loops and switch statements. There may be a situation when you need to come out of a loop without reaching its bottom. There may also be a situation when you want to skip a part of your code block and start the next iteration of the loop. To handle all such situations, Solidity provides **break** and **continue** statements. These statements are used to immediately come out of any loop or to start the next iteration of any loop respectively.

1. While Loop

```
pragma solidity ^0.5.0;  
  
contract SolidityTest {  
    uint storedData;  
  
    constructor() public{  
        storedData = 10;  
    }  
}
```

```
function getResult() public view returns(string memory){  
    uint a = 10;  
    uint b = 2;  
    uint result = a + b;  
    return integerToString(result);  
}  
  
function integerToString(uint _i) internal pure  
returns (string memory) {  
    if (_i == 0) {  
        return "0";  
    }  
    uint j = _i;  
    uint len;  
    while (j != 0) {  
        len++;  
        j /= 10;  
    }  
    bytes memory bstr = new bytes(len);  
    uint k = len - 1;  
    while (_i != 0) { // while loop  
        bstr[k--] = byte(uint8(48 + _i % 10));  
        _i /= 10;  
    }  
    return string(bstr);  
}
```

OUTPUT:

The screenshot shows the Remix Ethereum IDE interface. On the left, there's a sidebar with options like 'DEPLOY & RUN TRANSACTIONS', 'GAS LIMIT' set to 3000000, 'VALUE' set to 0 Wei, and a 'Deploy' button. Below that is a 'Transactions recorded' section showing a single transaction for 'getResult'. On the right, the main area displays the Solidity code for 'SolidityTest'. The code includes a constructor that initializes 'storedData' to 10, a public view function 'getResult' that returns the sum of 'a' and 'b', and an internal pure function 'integerToString' that converts integers to strings. A transaction log at the bottom shows a call from address 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to the contract, with a data value of 0xde2...92789.

```

pragma solidity ^0.5.0;

contract SolidityTest {
    uint storedData;
    constructor() public{
        storedData = 10;
    }
    function getResult() public view returns(string memory){
        uint a = 10;
        uint b = 2;
        uint result = a + b;
        return integerToString(result);
    }
    function integerToString(uint _i) internal pure
    returns (string memory) {
        if (_i == 0) {
            return "0";
        }
        uint j = _i;
        uint len;
        while (j != 0) {
            len++;
            j /= 10;
        }
        bytes memory bstr = new bytes(len);
        uint k = len - 1;
        while (_i != 0) { // while loop
            bstr[k] = byte(uint8(48 + _i % 10));
            _i /= 10;
        }
    }
}

```

2.Do-while loop:

```

pragma solidity ^0.5.0;

contract SolidityTest {
    uint storedData;
    constructor() public{
        storedData = 10;
    }
    function getResult() public view returns(string memory){
        uint a = 10;
        uint b = 2;
        uint result = a + b;
        return integerToString(result);
    }
    function integerToString(uint _i) internal pure
    returns (string memory) {
        if (_i == 0) {
            return "0";
        }

```

```

}

uint j = _i;

uint len;

while (j != 0) {

len++;

j /= 10;

}

bytes memory bstr = new bytes(len);

uint k = len - 1;

do {           // do while loop

bstr[k--] = byte(uint8(48 + _i % 10));

_i /= 10;

}

while (_i != 0);

return string(bstr);

}
}

```

OUTPUT:

The screenshot shows the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is open, showing deployment settings like GAS LIMIT (3,000,000), VALUE (0 Wei), and CONTRACT (SolidityTest - prac.sol). Below this, the 'Transactions recorded' section shows a single transaction for the contract 'SOLIDITYTEST AT 0XF8E...9FBEB'. On the right, the main workspace displays the Solidity source code for 'prac.sol'. The code defines a function `integerToString` that takes an integer `_i` and returns a string representation. It uses a do-while loop to repeatedly divide `_i` by 10 and store the result as a character in a byte array `bstr`. The bottom of the screen shows the EVM call trace for a recent call to the `getResult` function, which returned the string '0'.

```

function integerToString(uint _i) internal pure
    returns (string memory) {
    if (_i == 0) {
        return "0";
    }

    uint j = _i;
    uint len;

    while (j != 0) {
        len++;
        j /= 10;
    }

    bytes memory bstr = new bytes(len);
    uint k = len - 1;

    do {           // do while loop
        bstr[k--] = byte(uint8(48 + _i % 10));
        _i /= 10;
    }

    while (_i != 0);
    return string(bstr);
}

```

3.For Loop:

```
pragma solidity ^0.5.0;

contract SolidityTest {
    uint storedData;
    constructor() public{
        storedData = 10;
    }

    function getResult() public view returns(string memory){
        uint a = 10;
        uint b = 2;
        uint result = a + b;
        return integerToString(result);
    }

    function integerToString(uint _i) internal pure
    returns (string memory) {
        if (_i == 0) {
            return "0";
        }
        uint j=0;
        uint len;
        for (j = _i; j != 0; j /= 10) { //for loop example
            len++;
        }
        bytes memory bstr = new bytes(len);
        uint k = len - 1;
        while (_i != 0) {
            bstr[k--] = byte(uint8(48 + _i % 10));
        }
    }
}
```

```

_i /= 10;

}

return string(bstr);//access local variable

}}

```

OUTPUT:

The screenshot shows the Remix Ethereum IDE interface. On the left, there's a sidebar for 'DEPLOY & RUN TRANSACTIONS' with fields for GAS LIMIT (3000000), VALUE (0 Wei), and CONTRACT (SolidityTest - pracs.sol). Below it are buttons for Deploy, Publish to IPFS, At Address, and Load contract from Address. In the center, the Solidity code for 'pracs.sol' is displayed, showing the implementation of the 'integerToString' function. On the right, the transaction history shows a call to 'SolidityTest.getResult()' with a return value of '0xde2...92789'.

4.loop Control: (Break statement)

```

pragma solidity ^0.5.0;

contract SolidityTest {

    uint storedData;

    constructor() public{
        storedData = 10;
    }

    function getResult() public view returns(string memory){
        uint a = 1;
        uint b = 2;
        uint result = a + b;
    }
}

```

```
return integerToString(result);
}

function integerToString(uint _i) internal pure
returns (string memory) {

if (_i == 0) {
return "0";
}
uint j = _i;
uint len;

while (true) {
len++;
j /= 10;
if(j==0){
break; //using break statement
}
}

bytes memory bstr = new bytes(len);
uint k = len - 1;

while (_i != 0) {
bstr[k--] = byte(uint8(48 + _i % 10));
_i /= 10;
}

return string(bstr);
}
```

OUTPUT:

```

DEPLOY & RUN TRANSACTIONS
GAS LIMIT: 3000000
VALUE: 0 Wei
CONTRACT: SolidityTest - prac.sol
Deploy
Publish to IPFS
OR
At Address Load contract from Address
Transactions recorded: 5
Deployed Contracts
SOLIDITYTEST AT 0x7EF...8CB47 (ME)
 getResult
0: string: 3
Low level interactions
CALLDATA
Transact
https://solidity.readthedocs.io/en/v0.6.2/contracts.html#receive-ether-function

```

```

7 }
8     function getResult() public view returns(string memory){
9         uint a = 1;
10        uint b = 2;
11        uint result = a + b;
12        return integerToString(result);
13    }
14    function integerToString(uint _i) internal pure
15        returns (string memory) {
16
17        if (_i == 0) {
18            return "0";
19        }
20        uint j = -_i;
21        uint len;
22
23        while (true) {
24            len++;
25            j /= 10;
26            if(j==0){
27                break; //using break statement
28            }
29        }
30        bytes memory bstr = new bytes(len);
31        uint k = len - 1;
32
33        while (_i != 0) {
34            bstr[k--] = byte(uint8(48 + _i % 10));
35            _i /= 10;
36        }
37        return string(bstr);
38    }

```

CALL [call] from: 0x5B38Da6a701c568545dCfcB03Fc8875f56beddC4 to: SolidityTest.getResult() data: 0xde2...92789

(continue statement)

```

pragma solidity ^0.5.0;

contract SolidityTest {
    uint storedData;

    constructor() public{
        storedData = 10;
    }

    function getResult() public view returns(string memory){
        uint n = 1;
        uint sum = 0;

        while( n < 10){
            n++;
            if(n == 5){
                continue; // skip n in sum when it is 5.
            }

```

```
sum = sum + n;  
}  
  
return integerToString(sum);  
}  
  
function integerToString(uint _i) internal pure  
returns (string memory) {  
  
if (_i == 0) {  
return "0";  
}  
  
uint j = _i;  
uint len;  
  
while (true) {  
len++;  
j /= 10;  
if(j==0){  
break; //using break statement  
}  
}  
  
bytes memory bstr = new bytes(len);  
uint k = len - 1;  
  
while (_i != 0) {  
bstr[k--] = byte(uint8(48 + _i % 10));  
_i /= 10;  
}  
  
return string(bstr);  
}
```

```
}
```

OUTPUT:

```

function integerToString(uint _i) internal pure
    returns (string memory) {
    if (_i == 0) {
        return "0";
    }
    uint j = _i;
    uint len;

    while (true) {
        len++;
        j /= 10;
        if(j==0){
            break; //using break statement
        }
    }
    bytes memory bstr = new bytes(len);
    uint k = len - 1;

    while (_i != 0) {
        bstr[k--] = byte(uint8(48 + _i % 10));
        _i /= 10;
    }
    return string(bstr);
}

```

D)Decision Making:

While writing a program, there may be a situation when you need to adopt one out of a given set of paths. In such cases, you need to use conditional statements that allow your program to make correct decisions and perform right actions. Solidity supports conditional statements which are used to perform different actions based on different conditions. Here we will explain the **if..else** statement.

1.if statement: The **if** statement is the fundamental control statement that allows Solidity to make decisions and execute statements conditionally.

```

pragma solidity ^0.5.0;

contract SolidityTest {
    uint storedData;
    constructor() public {
        storedData = 10;
    }
}
```

```
}

function getResult() public view returns(string memory){
    uint a = 1;
    uint b = 2;
    uint result = a + b;
    return integerToString(result);
}

function integerToString(uint _i) internal pure
returns (string memory) {
    if (_i == 0) { // if statement
        return "0";
    }
    uint j = _i;
    uint len;

    while (j != 0) {
        len++;
        j /= 10;
    }
    bytes memory bstr = new bytes(len);
    uint k = len - 1;

    while (_i != 0) {
        bstr[k--] = byte(uint8(48 + _i % 10));
        _i /= 10;
    }
    return string(bstr); //access local variable
}
```

OUTPUT:

```

DEPLOY & RUN TRANSACTIONS
GAS LIMIT: 3000000
VALUE: 0 Wei
CONTRACT: SolidityTest - prac.sol
Deploy
Publish to IPFS OR At Address Load contract from Address
Transactions recorded 7 Deployed Contracts
SOLIDITYTEST AT 0X358...D5EE3 (IM)
getResult 0: string: 3
Low level interactions CALLDATA
Transact

```

```

4 uint storedData;
5 constructor() public {
6     storedData = 10;
7 }
8 function getResult() public view returns(string memory){
9     uint a = 1;
10    uint b = 2;
11    uint result = a + b;
12    return integerToString(result);
13 }
14 function integerToString(uint _i) internal pure
15     returns (string memory) {
16     if (_i == 0) { // if statement
17         return "0";
18     }
19     uint j = _i;
20     uint len;
21
22     while (j != 0) {
23         len++;
24         j /= 10;
25     }
26     bytes memory bstr = new bytes(len);
27     uint k = len - 1;
28
29     while (_i != 0) {
30         bstr[k--] = byte(uint8(48 + _i % 10));
31         _i /= 10;
32     }
33     return string(bstr); //access local variable
34 }
35

```

[call] from: 0x5B3...eddC4 to: SolidityTest.getResult() data: 0xde2...92789

2.if-else statement: The 'if...else' statement is the next form of control statement that allows Solidity to execute statements in a more controlled way.

```

pragma solidity ^0.5.0;
// Creating a contract
contract Types {
// Declaring state variables
uint i = 10;
bool even;
// Defining function to
// demonstrate the use of
// 'if...else statement'
function decision_making()
public payable returns(bool){
if (i%2 == 0){
even = true;
}
else{
even = false;
}
return even;
}
}

```

OUTPUT:

```

1 pragma solidity ^0.5.0;
2
3 // Creating a contract
4 contract Types {
5     // Declaring state variables
6     uint i = 10;
7     bool even;
8
9     // Defining function to
10    // demonstrate the use
11    // of 'if...else if...else
12    // statement'
13    function decision_making()
14        public payable returns(bool){
15        if (i%2 == 0){
16            even = true;
17        }
18        else{
19            even = false;
20        }
21        return even;
22    }
23 }
```

3.if-else..if statement: The **if...else if...** statement is an advanced form of **if...else** that allows Solidity to make a correct decision out of several conditions.

pragma solidity ^0.5.0;

```

// Creating a contract
contract Types {
// Declaring state variables
uint i = 12;
string result;
// Defining function to
// demonstrate the use
// of 'if...else if...else
// statement'
function decision_making (
) public returns(string memory){
if(i<10){
result = "less than 10";
}
else if(i == 10){
result = "equal to 10";
}
else{
result = "greater than 10";
}
```

```

    }
    return result;
}
}

```

OUTPUT:

```

DEPLOY & RUN TRANSACTIONS
GAS LIMIT: 3000000
VALUE: 0 Wei
CONTRACT: Types - prac.sol
Deploy OR At Address Load contract from Address
Transactions recorded: 4 Deployed Contracts
TYPES AT 0xf8e...9f8e8 (MEMORY)
decision_making()
Low level interactions CALLDATA Transaction
[vm] from: 0x5b3...eddC4 to: types.decision_making() 0xf8e...9f8e8 value: 0 wei data: 0x887...3af21 logs: 0 hash: 0xac2...6e37e

```

String:

// Solidity program to demonstrate
// how to create a contract

```

pragma solidity ^0.4.23;

// Creating a contract
contract Test {
// Declaring variable
string str;

// Defining a constructor
constructor(string str_in){
str = str_in;
}

// Defining a function to
// return value of variable 'str'
function str_out() public view returns(string memory){
return str;
}
}

```

Note: after deploy it asked u to enter string then enter string over there and then see the output after clicking on str_out button

OUTPUT:

```
// Solidity program to demonstrate
// how to create a contract
pragma solidity ^0.4.23;

contract Test {
    string str;

    constructor(string str_in) {
        str = str_in;
    }

    // Defining a function to
    // return value of variable 'str'
    function str_out() public view returns(string memory){
        return str;
    }
}
```

A) String:

Solidity supports String literal using both double quote ("") and single quote (''). It provides string as a data type to declare a variable of type String.(Int to str)

pragma solidity ^0.5.0;

```
contract SolidityTest {
constructor() public{
}

function getResult() public view returns(string memory){
uint a = 1;
uint b = 2;
uint result = a + b;
return integerToString(result);
}

function integerToString(uint _i) internal pure
```

```

returns (string memory) {

if (_i == 0) {

return "0";

}

uint j = _i;

uint len;

while (j != 0) {

len++;

j /= 10;

}

bytes memory bstr = new bytes(len);

uint k = len - 1;

while (_i != 0) {

bstr[k--] = byte(uint8(48 + _i % 10));

_i /= 10;

}

return string(bstr);

}
}

```

OUTPUT:

B)Array:

Array is a data structure, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

```
// Solidity program to demonstrate  
// accessing elements of an array
```

```
pragma solidity ^0.5.0;  
  
// Creating a contract  
contract Types {  
  
    // Declaring an array  
    uint[6] data;  
    uint x;  
  
    // Defining function to  
    // assign values to array  
    function array_example() public returns (uint[6] memory)  
    {  
  
        data = [uint(10), 20, 30, 40, 50, 60];  
    }  
    function result() public view returns(uint[6] memory){  
        return data;  
    }  
    // Defining function to access  
    // values from the array  
    // from a specific index  
    function array_element() public view returns (uint){  
        uint x = data[2];  
        return x;  
    }  
}
```

OUTPUT:

```
// Solidity program to demonstrate
// accessing elements of an array
pragma solidity ^0.5.0;

// Creating a contract
contract Types {
    // Declaring an array
    uint[6] data;
    uint x;

    // Defining function to
    // assign values to array
    function array_example() public returns (uint[6] memory) {
        data = [uint(10), 20, 30, 40, 50, 60];
    }

    // Defining function to access
    // values from the array
    // from a specific index
    function array_element() public view returns (uint){
        uint x = data[2];
        return x;
    }
}
```

C)Enums:

Enums restrict a variable to have one of only a few predefined values. The values in this enumerated list are called enums. With the use of enums it is possible to reduce the number of bugs in your code.

```
// Solidity program to demonstrate
// how to use 'enumerator'
```

```
pragma solidity ^0.5.0;

// Creating a contract
contract Types {

    // Creating an enumerator
    enum week_days {
        Monday,
        Tuesday,
        Wednesday,
        Thursday,
        Friday,
        Saturday,
        Sunday
    }

    // Declaring variables of
```

```

// type enumerator
week_days week;

week_days choice;

// Setting a default value
week_days constant default_value
= week_days.Sunday;

// Defining a function to
// set value of choice
function set_value() public {
choice = week_days.Thursday;
}

// Defining a function to
// return value of choice
function get_choice(
) public view returns (week_days) {
return choice;
}

// Defining function to
// return default value
function getdefaultValue(
) public pure returns(week_days) {
return default_value;
}
}

```

OUTPUT:

The screenshot shows the Remix Ethereum IDE interface. In the top navigation bar, the URL is `remix.ethereum.org/optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d10bba13.js&language=Solidity`. The main area is titled "DEPLOY & RUN TRANSACTIONS". A dropdown menu shows "Contract: Types - prac.sol". Below it, there are buttons for "Deploy" and "Publish to IPFS". Under "Transactions recorded", there are three buttons: "set_value", "get_choice", and "getdefaultValue". The "getdefaultValue" button is highlighted in blue. The "Deployed Contracts" section shows a list with one item: "set_value" (0 Wei). At the bottom, there is a "Low level interactions" section with a "CALLDATA" field and a "Transact" button.

```

19 // Declaring variables of
20 // type enumerator
21 week_days week;
22
23 week_days choice;
24
25 // Setting a default value
26 week_days constant default_value
27 = week_days.Sunday;
28
29
30 // Defining a function to
31 // set value of choice
32 function set_value() public {
33 choice = week_days.Thursday;
34 }
35
36 // Defining a function to
37 // return value of choice
38 function get_choice(
39 ) public view returns (week_days) {
40 return choice;
41 }
42
43 // Defining function to
44 // return default value
45 function getdefaultValue(
46 ) public pure returns (week_days) {
47 return default_value;
48 }

```

At the bottom of the screen, a transaction log is visible:

```

0x5B38Da6a701c568545dCfc03Fc875f56beddC4 to: Types.getdefaultValue() data: 0xbd7...2e4fc

```

D)Structure:

Struct types are used to represent a record.

```
pragma solidity ^0.5.0;
```

```
contract test {
    struct Book {
        string title;
        string author;
        uint book_id;
    }
    Book book;

    function setBook() public {
        book = Book('Learn Java', 'TP', 1);
    }
    function getBookId() public view returns (uint) {
        return book.book_id;
    }
}
```

OUTPUT:

The screenshot shows the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is visible, showing deployment options like 'Deploy' and 'Publish to IPFS'. The main central area displays the Solidity code for the 'test' contract. Below the code, the 'Transactions recorded' section shows two transactions: 'setBook' and 'getBookId'. The 'Low level interactions' section at the bottom shows a single call to the 'getBookId' function. The bottom right corner has a 'Debug' button.

```
pragma solidity ^0.5.0;

contract test {
    struct Book {
        string title;
        string author;
        uint book_id;
    }
    Book book;

    function setBook() public {
        book = Book('Learn Java', 'TP', 1);
    }
    function getBookId() public view returns (uint) {
        return book.book_id;
    }
}
```

E)Mappings:

Mapping is a reference type as arrays and structs. Following is the syntax to declare a mapping type.

mapping(_KeyType => _ValueType) where ,

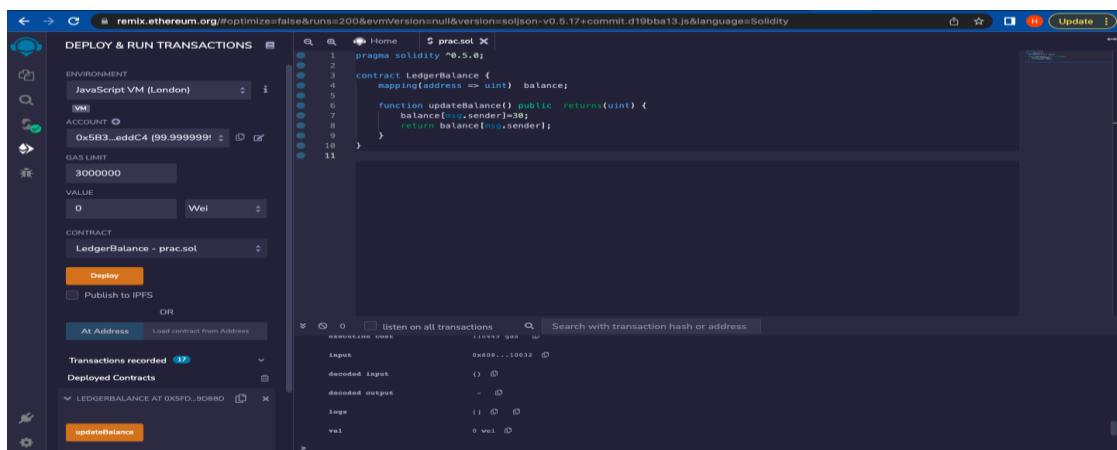
_KeyType – can be any built-in types plus bytes and string. No reference type or complex objects are allowed.

_ValueType – can be any type.

pragma solidity ^0.5.0;

```
contract LedgerBalance {
    mapping(address => uint) balance;
    function updateBalance() public returns(uint) {
        balance[msg.sender]=30;
        return balance[msg.sender];
    }
}
```

OUTPUT:



Mapping program for String.

```
pragma solidity ^0.5.0;
contract LedgerBalance {
    mapping(address => string) name;
    function updateBalance() public returns(string memory){
        name[msg.sender] = "Mrunali";
        return name[msg.sender];
    }
    function printsender() public view returns(address) {
        return msg.sender;
    }
}
```

{
}

{

OUTPUT:

The screenshot shows the Remix Ethereum IDE interface. On the left, there's a sidebar with icons for file operations, search, and deployment. The main area has tabs for "Home" and "prac.sol". The code editor contains the following Solidity code:

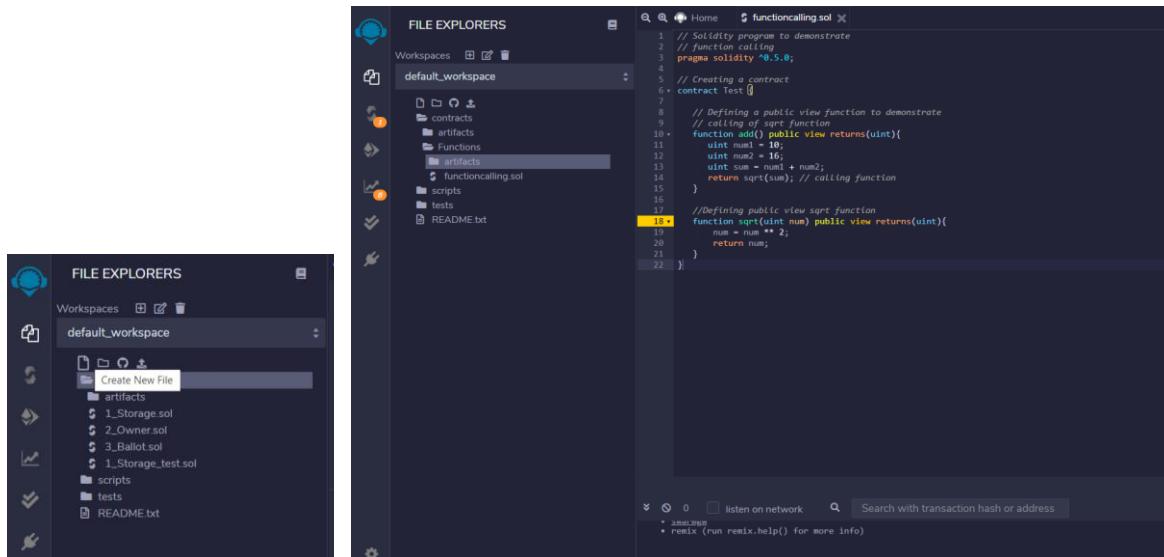
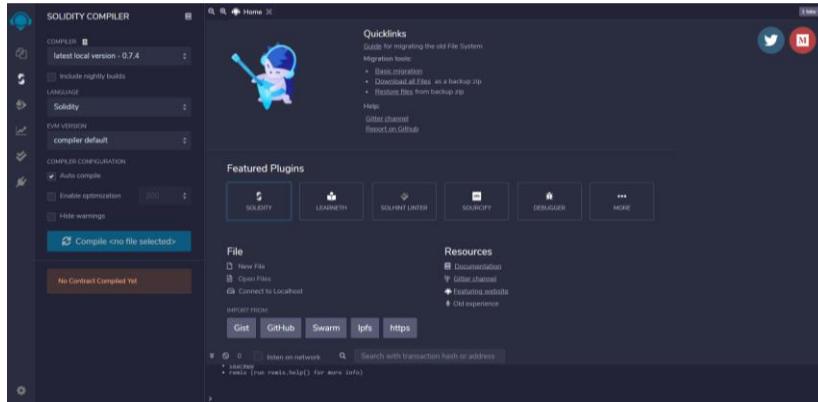
```
1 pragma solidity >=0.5.0;
2
3 contract Ledgerbalance {
4     mapping(address => string) name;
5
6     function updatebalance() public returns(string memory){
7         name[msg.sender] = "Mrunal";
8         return name[msg.sender];
9     }
10    function printsender() public view returns(address) {
11        return msg.sender;
12    }
13 }
14
```

Below the code, there are sections for "DEPLOY & RUN TRANSACTIONS" and "Deployed Contracts". Under "Deployed Contracts", it shows a deployed contract named "LEDGERBALANCE AT 0x789...B6ACE" with two functions listed: "updatebalance" and "printsender". It also shows the address of the deployed contract: 0x5B38Da6a701c568545dCfcB03F08875f56beddC4. At the bottom, there's a "Low level interactions" section with a "CALLDATA" tab and a "Transact" button.

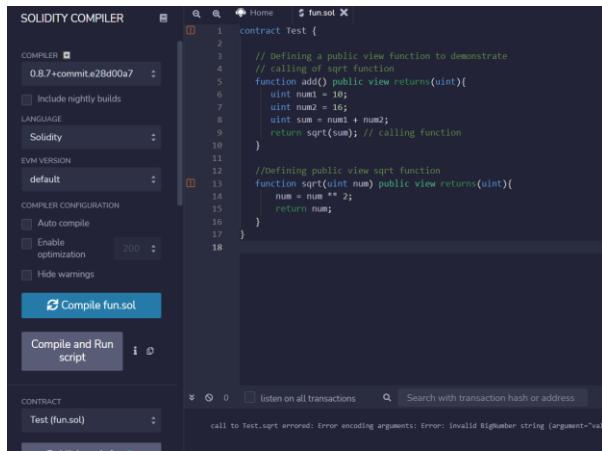
Practical 3B

Aim:

Implement and demonstrate the use of the following in Solidity: Functions, Function Modifiers, View functions, Pure Functions, Fallback Function, Function Overloading, Mathematical functions, Cryptographic functions.



- Step 2 – Under Compile Tab, click Start to Compile button.



- Step 3 – Under Run Tab, click Deploy button.
- Step 4 – Click add and sqrt Button to display the result.

Functions

- Function Calling

Code

```
// Solidity program to demonstrate
// function calling
pragma solidity ^0.5.0;

// Creating a contract
contract Test {

    // Defining a public view function to demonstrate
    // calling of sqrt function
    function add() public view returns(uint){
        uint num1 = 10;
        uint num2 = 16;
        uint sum = num1 + num2;
        return sqrt(sum); // calling function
    }

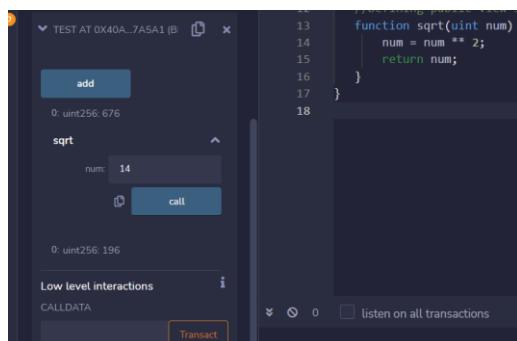
    //Defining public view sqrt function
    function sqrt(uint num) public view returns(uint){
        num = num ** 2;
        return num;
    }
}
```

```

        uint num2 = 16;
        uint sum = num1 + num2;
        return sqrt(sum); // calling function
    }

//Defining public view sqrt function
function sqrt(uint num) public view returns(uint){
    num = num ** 2;
    return num;
}

```

Output:

- **Return statement**

Code

```

// Solidity program to demonstrate
// return statements
pragma solidity ^0.5.0;

// Creating a contract
contract Test {

    // Defining a public view function to
    // demonstrate return statement
    function return_example() public view returns(uint, uint, uint, string memory){
        uint num1 = 26;
        uint num2 = 24;
        uint sum = num1 + num2;
        uint prod = num1 * num2;
        uint diff = num2 - num1;
        string memory msg = "Multiple return values";
        return (sum, prod, diff, msg);
    }
}

```

```
}
```

Output

```

15     uint diff = num2 - num1;
16     string memory msg = "Multiple return values";
17     return (sum, prod, diff, msg);
18   }
19 }
20

```

Low level interactions

CALldata

Transact

CALL [call] from: 0x8F3c43ff999573d4ffFeA15

Function Modifiers

Code

```

pragma solidity ^0.5.0;
contract Owner {
    address owner;
    constructor() public {
        owner = msg.sender;
    }
    modifier onlyOwner {
        require(msg.sender == owner);
        _;
    }
    modifier costs(uint price) {
        if (msg.value >= price) {
            _;
        }
    }
}
contract Register is Owner {
    mapping (address => bool) registeredAddresses;
    uint price;
    constructor(uint initialPrice) public { price = initialPrice; }

    function register() public payable costs(price) {
        registeredAddresses[msg.sender] = true;
    }
    function changePrice(uint _price) public onlyOwner {
        price = _price;
    }
}

```

[View Functions](#)**Code**

```
// Solidity program to
// demonstrate view
// functions
pragma solidity ^0.5.0;

// Defining a contract
contract Test {

    // Declaring state
    // variables
    uint num1 = 12;
    uint num2 = 14;

    // Defining view function to
    // calculate product and sum
    // of 2 numbers
    function getResult(
    ) public view returns(
        uint product, uint sum){
        uint num1 = 5;
        uint num2 = 8;
        product = num1 * num2;
        sum = num1 + num2;
    }
}
```

Output:

The screenshot shows a blockchain application interface. On the right, a sidebar displays a list of transactions with details like recipient, value, data, logs, and hash. One transaction is highlighted with a green checkmark, indicating it was successful. On the left, a main panel shows a contract named 'TEST AT 0xE26...95457 (BL)'. It has a 'getResult' button. Below it, two results are listed: '0: uint256: product 40' and '1: uint256: sum 13'. At the bottom, there's a 'Low level interactions' section and a 'Transact' button.

Pure Functions

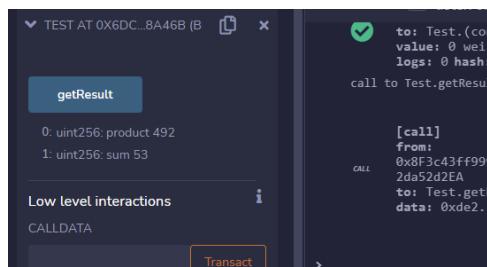
Code

```
// Solidity program to
// demonstrate pure functions
pragma solidity ^0.5.0;

// Defining a contract
contract Test {

    // Defining pure function to
    // calculate product and sum
    // of 2 numbers
    function getResult(
    ) public pure returns(
        uint product, uint sum){
        uint num1 = 12;
        uint num2 = 41;
        product = num1 * num2;
        sum = num1 + num2;
    }
}
```

Output:



Fallback Functions

Code

```
pragma solidity ^0.4.0;

// Creating a contract
contract GeeksForGeeks {
    // Declaring the state variable
    uint x;

    // Mapping of addresses to their balances
}
```

```
mapping(address => uint) balance;

// Creating a constructor
constructor() public
{
    // Set x to default
    // value of 10
    x=10;

}

// Creating a function
function SetX(uint _x) public returns(bool)
{
    // Set x to the
    // value sent
    x=_x;
    return true;
}

// This fallback function
// will keep all the Ether
function() public payable
{
    balance[msg.sender] += msg.value;
}
}

// Creating the sender contract
contract Sender
{
function transfer() public payable
{
    // Address of GeeksForGeeks contract
    address _receiver =
        0xbcc0185441de06F0452D45AEd6Ad8b98017796fb;

    // Transfers 100 Eth to above contract
    _receiver.transfer(100);
}
}
```

Output:

The screenshot shows the Truffle UI interface. On the left, there's a sidebar for 'CONTRACT' with options like 'Deploy', 'Publish to IPFS', and 'At Address'. Below that is a section for 'Transactions recorded' and 'Deployed Contracts'. A dropdown shows 'GEEKSFORGEeks AT 0xD91...39138'. On the right, detailed transaction information is displayed for a successful transaction:

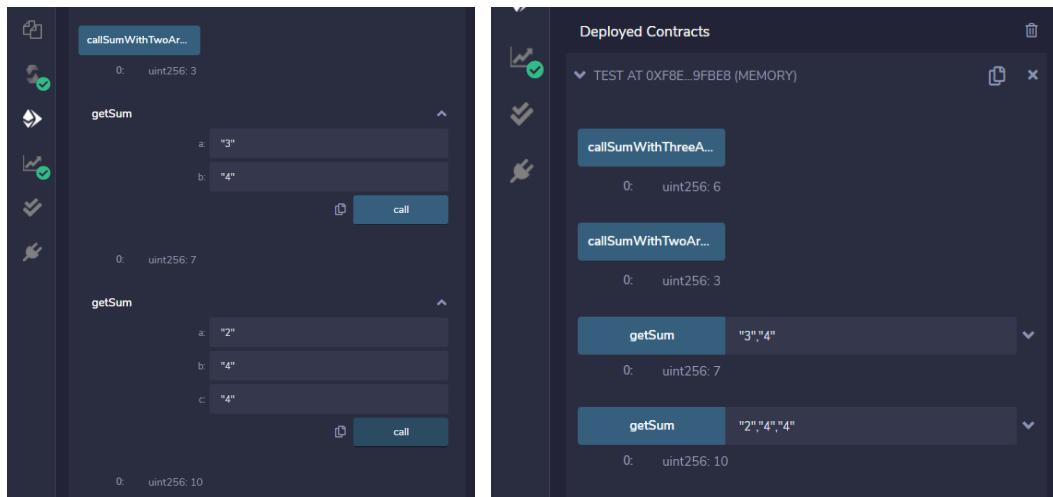
- status: true Transaction mined and execution succeed
- transaction hash: 0xf16c571cfbf3d359010fe699da770da6cc57780cd2fe01b35b3ad335c70e
- contract address: 0xd9145CCE5020386f254917e481e944e9943f39138
- from: 0x5B38D6a701c5605450CfcB83Fc8875f56beddC4
- to: GeeksForGeeks.(constructor)
- gas: 3000000 gas
- transaction cost: 145319 gas
- execution cost: 74119 gas
- hash: 0xf16c571cfbf3d359010fe699da770da6cc57780cd2fe01b35b3ad335c70e
- input: 0x600...90029
- decoded input: []
- decoded output: -
- logs: []
- value: 0 wei

Function Overloading**Code**

```
pragma solidity ^0.5.0;

contract Test {
    function getSum(uint a, uint b) public pure returns(uint){
        return a + b;
    }
    function getSum(uint a, uint b, uint c) public pure returns(uint){
        return a + b + c;
    }
    function callSumWithTwoArguments() public pure returns(uint){
        return getSum(1,2);
    }
    function callSumWithThreeArguments() public pure returns(uint){
        return getSum(1,2,3);
    }
}
```

Output:



Mathematical Functions

- Addition:

Code

```
// Solidity program to
// demonstrate addition
pragma solidity 0.6.6;
contract gfgMathPlus
{
    // Declaring the state
    // variables
    uint firstNo ;
    uint secondNo ;

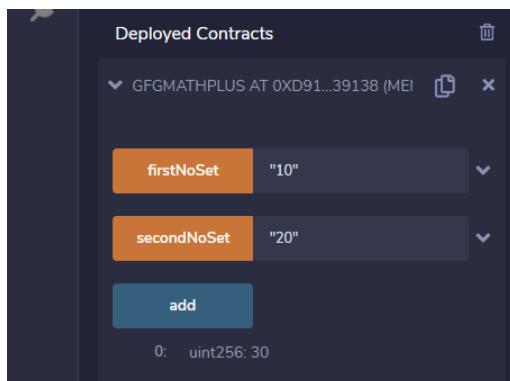
    // Defining the function
    // to set the value of the
    // first variable
    function firstNoSet(uint x) public
    {
        firstNo = x;
    }

    // Defining the function
    // to set the value of the
    // second variable
    function secondNoSet(uint y) public
    {
        secondNo = y;
    }

    // Defining the function
    // to get the sum of the
    // first and second variable
    function sum() public view returns (uint)
    {
        return firstNo + secondNo;
    }
}
```

```
// to add the two variables
function add() view public returns (uint)
{
    uint Sum = firstNo + secondNo ;

    // Sum of two variables
    return Sum;
}
```

Output:

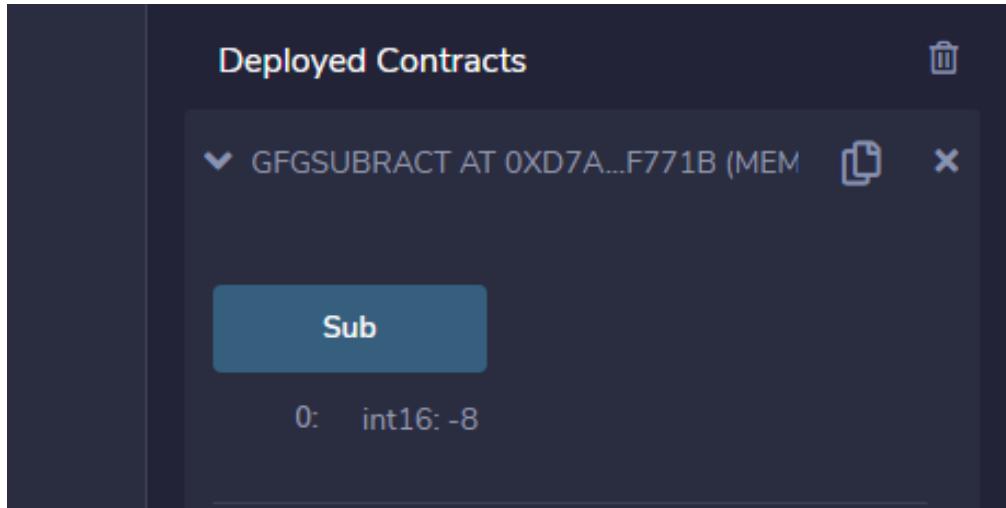
- **Subtraction:**

Code

```
// Solidity program to
// demonstrate the subtraction
pragma solidity 0.6.6;
contract gfgSubtract
{
    // Initializing the
    // state variables
    int16 firstNo=2 ;
    int16 secondNo=10;

    // Defining a function
    // to subtract two numbers
    function Sub() view public returns (int16)
    {
        int16 ans = firstNo - secondNo ;

        // Difference amount
        return ans;
    }
}
```

Output:

- **Multiplication**

Code

```
pragma solidity 0.6.6;
contract gfgMultiply {
    int128 firstNo ;
    int128 secondNo ;
    function firstNoSet(int128 x) public {
        firstNo = x; }
    function secondNoSet(int128 y) public {
        secondNo = y; }
    function multiply() view public returns (int128) {
        int128 answer = firstNo * secondNo ;
        return answer; }}
```

Output:

- Division:

Code

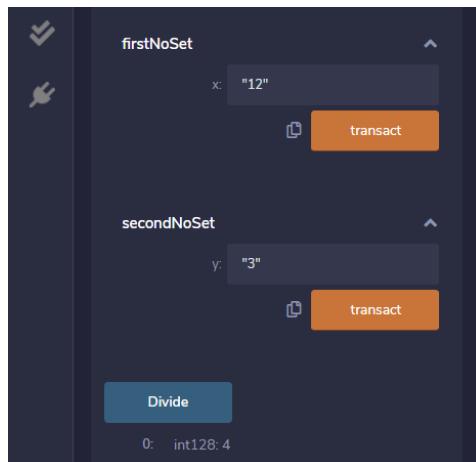
```
// Solidity program to
// demonstrate the
// division operation
pragma solidity 0.6.6;

// Creating a contract
contract gfgDivide
{
    // Declaring the
    // state variables
    int128 firstNo ;
    int128 secondNo ;

    // Defining a function
    // to set the value of
    // the first variable
    function firstNoSet(int64 x) public
    {
        firstNo = x;
    }

    // Defining function
    // to set the value of
    // the second variable
    function secondNoSet(int64 y) public
    {
        secondNo = y;
    }

    // Defining function to
    // return the result
    function Divide() view public returns (int128)
    {
        int128 answer = firstNo / secondNo ;
        return answer;
    }
}
```

Output:

- **Modulus:**

Code

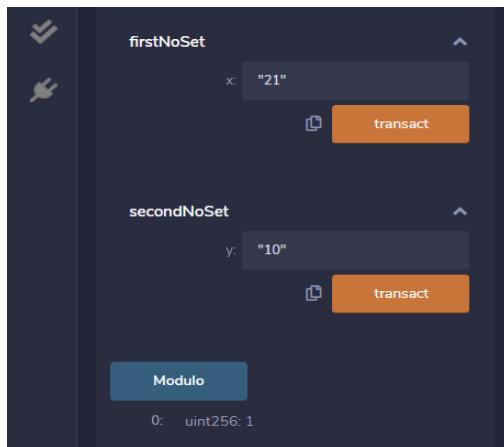
```
// Solidity program to
// demonstrate the
// Modulus operation
pragma solidity ^0.6.6;

// Creating a contract
contract gfgModulo
{
    // Declaring state variables
    uint firstNo ;
    uint secondNo ;

    // Defining a function
    // to set the value of
    // the first variable
    function firstNoSet(uint x) public
    {
        firstNo = x;
    }

    // Defining a function
    // to set the value of
    // the second variable
    function secondNoSet(uint y) public
    {
        secondNo = y;
    }
}
```

```
// Defining a function to return
// the modulus value
function Modulo() view public returns (uint)
{
    uint answer = firstNo % secondNo ;
    return answer;
}
```

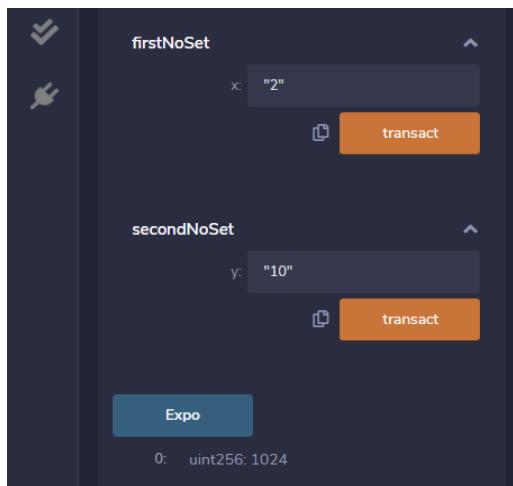
Output:

- **Exponentiation:**

Code

```
// Solidity program to
// demonstrate the
// exponentiation operation
pragma solidity ^0.6.6;
// Creating a contract
contract gfgExpo
{
    // Declaring the state
    // variables
    uint16 firstNo ;
    uint16 secondNo ;
    // Defining the first function
    // to set the value of
    // first variable
    function firstNoSet(uint16 x) public
    {
        firstNo = x;
    }
```

```
// Defining the function
// to set the value of
// the second variable
function secondNoSet(uint16 y) public
{secondNo = y;}
// Defining the function to
// calculate the exponent
function Expo() view public returns (uint256) {
    uint256 answer = firstNo ** secondNo ;
    return answer; }
```

Output:**Cryptographic Functions**

Solidity provides inbuilt cryptographic functions as well. Following are important methods –

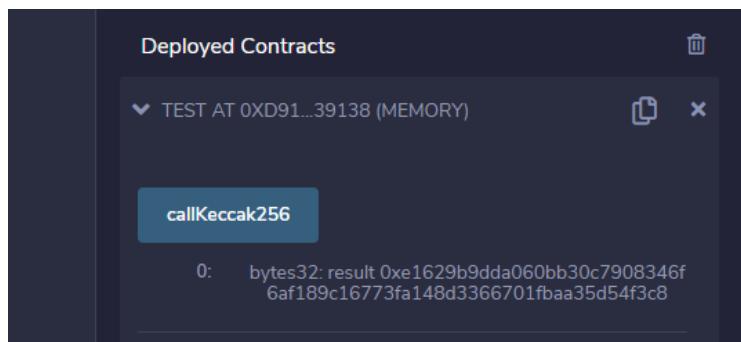
- keccak256(bytes memory) returns (bytes32) – computes the Keccak-256 hash of the input.
- sha256(bytes memory) returns (bytes32) – computes the SHA-256 hash of the input.
- ripemd160(bytes memory) returns (bytes20) – compute RIPEMD-160 hash of the input.
- sha256(bytes memory) returns (bytes32) – computes the SHA-256 hash of the input.
- ecrecover(bytes32 hash, uint8 v, bytes32 r, bytes32 s) returns (address) – recover the address associated with the public key from elliptic curve signature or return zero on error. The function parameters correspond to ECDSA values of the signature: r - first 32 bytes of signature; s: second 32 bytes of signature; v: final 1 byte of signature. This method returns an address.

Code

```
pragma solidity ^0.5.0;
contract Test {
```

```
function callKeccak256() public pure returns(bytes32 result){  
    return keccak256("ABC");  
}  
}
```

Output:



Practical 4A

Aim:

Implement and demonstrate the use of the following in Solidity.

Withdrawal Pattern, Restricted Access

Withdrawal pattern ensures that direct transfer call is not made which poses a security threat.

<https://www.wealdtech.com/articles/why-use-the-withdrawal-pattern/>

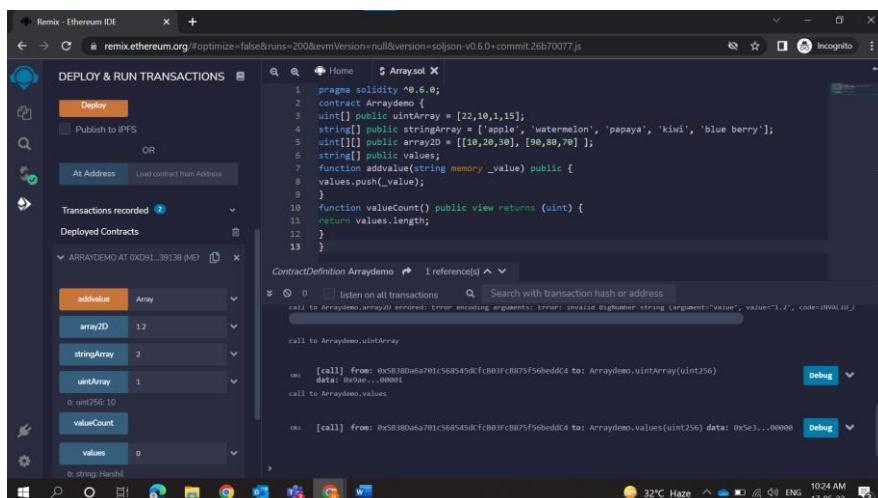
The withdrawal pattern places the responsibility for claiming funds, on the recipient of the funds: the recipient has to send a transaction to withdraw and obtain their funds.

This can simplify a smart contract that is sending funds to recipients, because the contract does not have to handle the cases of what to do if sending the funds fail. A smart contract does not know whether sending the funds failed due to an actual error, or whether the recipient is a malicious smart contract that deliberately refuses to accept the funds.

CODE:

```
pragma solidity ^0.6.0;
contract Arraydemo {
    uint[] public uintArray = [22,10,1,15];
    string[] public stringArray = ['apple', 'watermelon', 'papaya', 'kiwi', 'blue berry'];
    uint[][] public array2D = [[10,20,30], [90,80,70] ];
    string[] public values;
    function addvalue(string memory _value) public {
        values.push(_value);
    }
    function valueCount() public view returns (uint) {
        return values.length;
    }
}
```

OUTPUT:

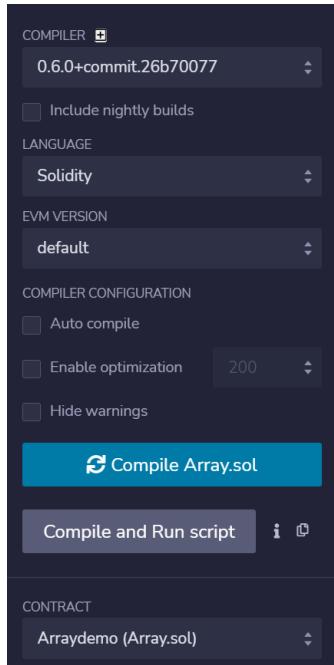


Restricted Access to a Contract is a common practice. By Default, a contract state is read-only unless it is specified as public.

We can restrict who can modify the contract's state or call a contract's functions using modifiers.

We will create and use multiple modifiers as explained below.

- **onlyBy** – once used on a function then only the mentioned caller can call this function.
- **onlyAfter** – once used on a function then that function can be called after certain time period.
- **costs** – once used on a function then caller can call this function only if certain value is provided.



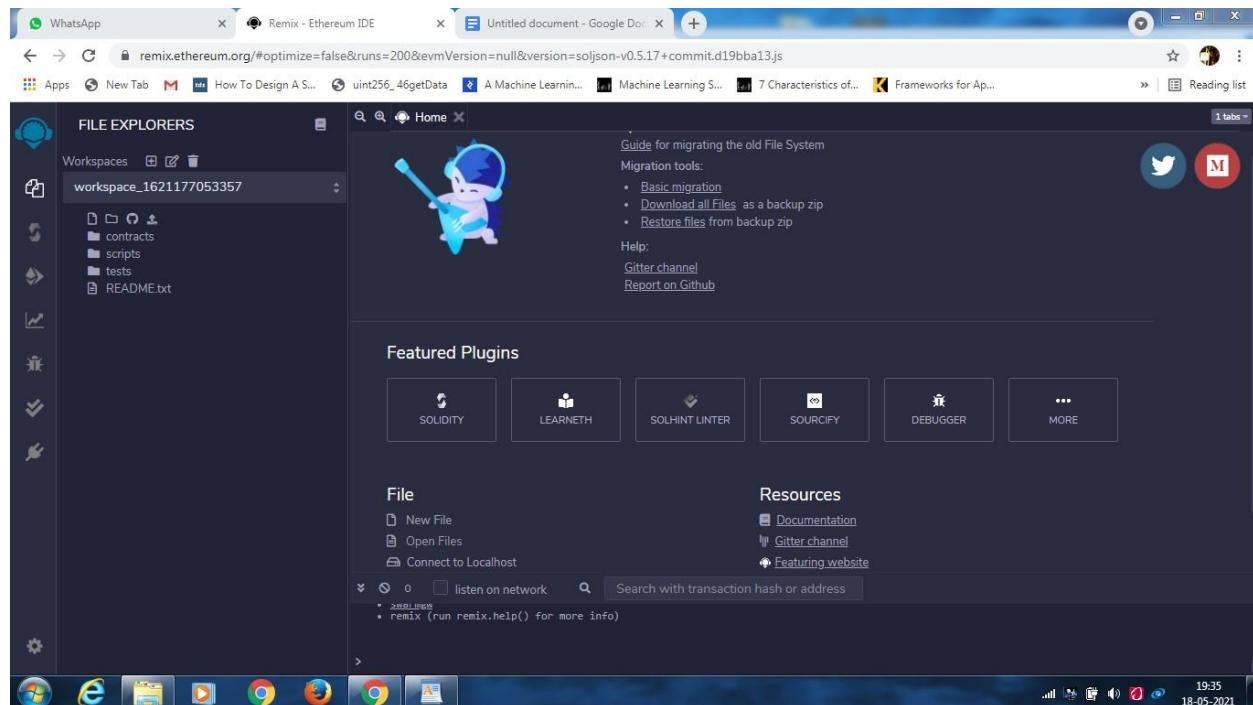
PRACTICAL 4B

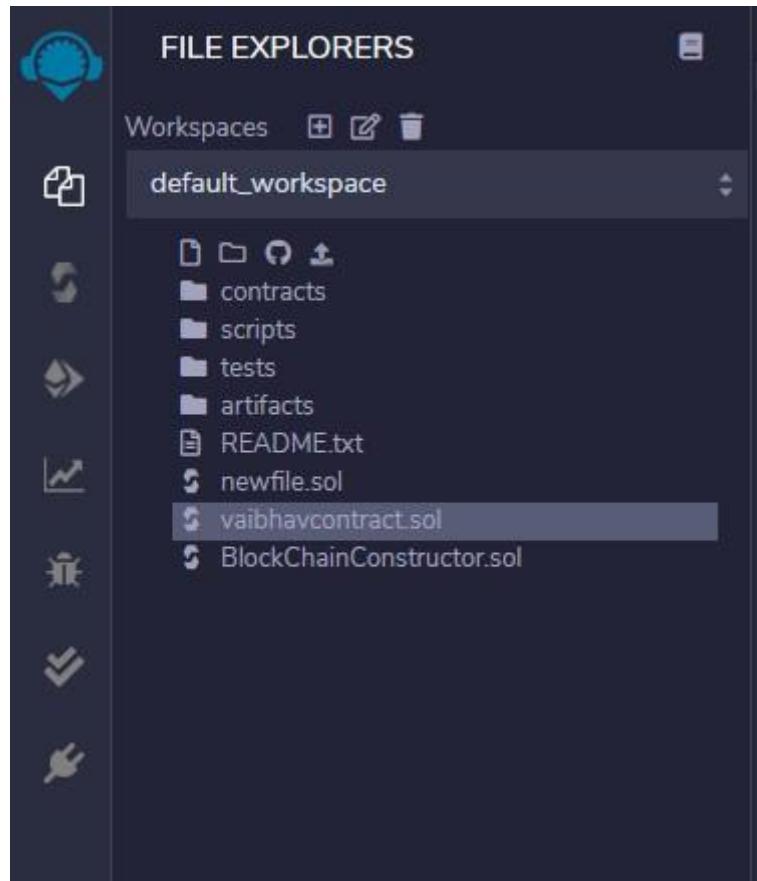
Aim: Contracts, Inheritance, Constructors, Abstract Contracts, Interfaces.

i) Implement a smart contract.

Steps:

Step 1:- create a file in online IDE remix -ethereum using solditiy programming language and name the file in default workspace.

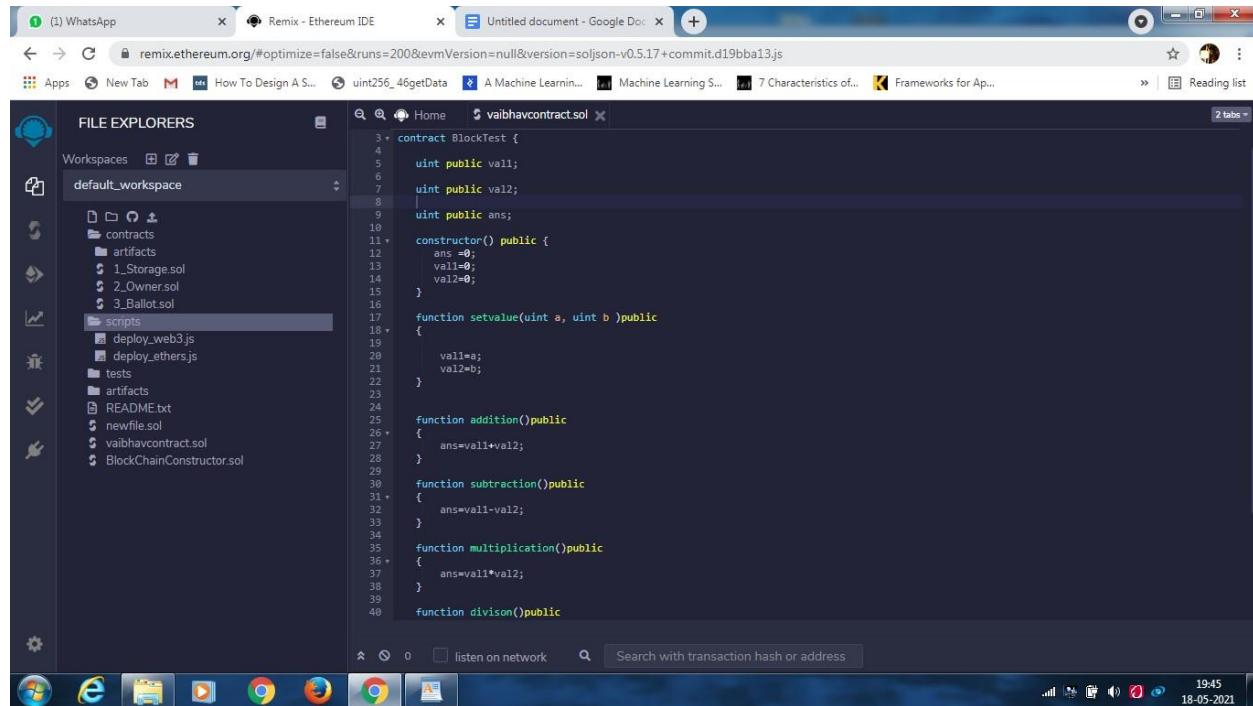




Step 2:- here we create a contract for calculator and name the contract as Blocktest and having a definition of heading as **pragma solidity ^0.5.0;**

```
1 pragma solidity ^0.5.0;
2
3 contract BlockTest {
4
5     uint public val1;
6
7     uint public val2;
8
9     uint public ans;
10 }
```

Step 3:- In this language the contract in solidity is same work as class in C++ , and inside this contract will have state variables with modifiers, constructor and function with modifiers (internal , external,private,public)



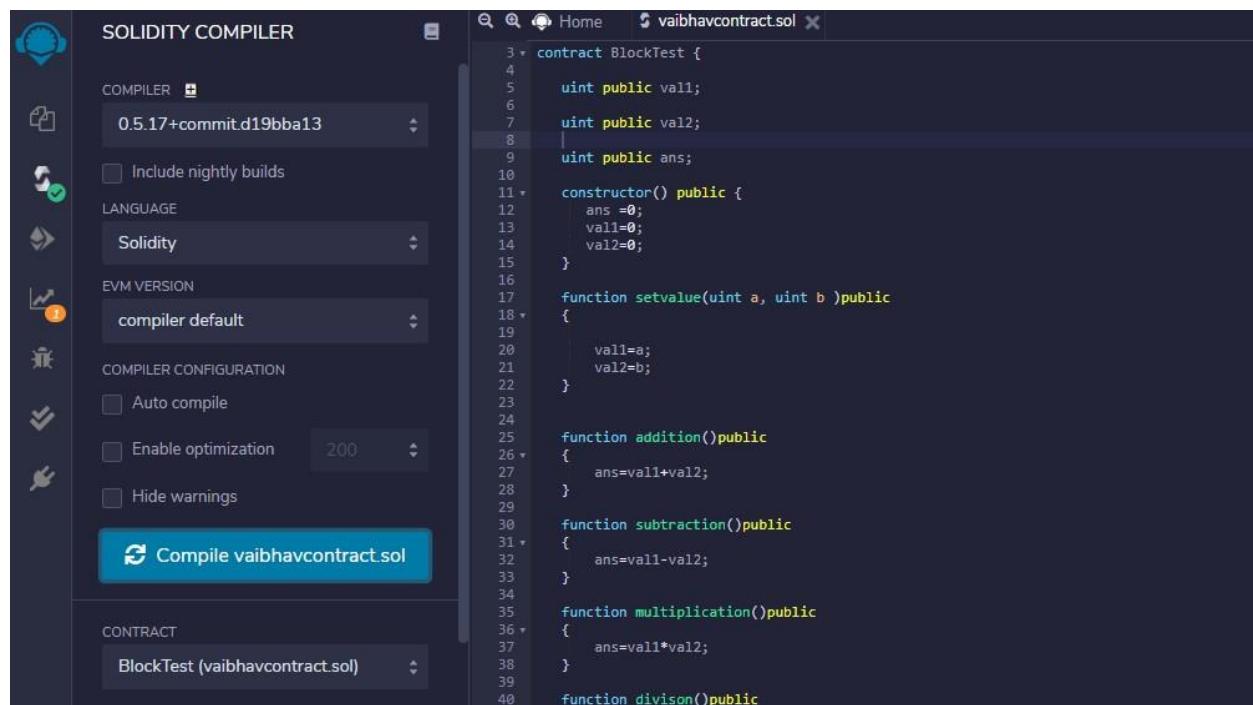
The screenshot shows the Remix Ethereum IDE interface. On the left, the FILE EXPLORERS panel displays a workspace named 'default_workspace' containing contracts like 1_Storage.sol, 2_Owner.sol, 3_Ballot.sol, and scripts like deploy.web3.js and deploy.ethers.js. The central code editor window shows the Solidity code for the 'BlockTest' contract:

```

3 + contract BlockTest {
4     uint public val1;
5
6     uint public val2;
7
8     uint public ans;
9
10    constructor() public {
11        ans = 0;
12        val1 = 0;
13        val2 = 0;
14    }
15
16    function setvalue(uint a, uint b )public
17    {
18        val1=a;
19        val2=b;
20    }
21
22
23    function addition()public
24    {
25        ans=val1+val2;
26    }
27
28    function subtraction()public
29    {
30        ans=val1-val2;
31    }
32
33    function multiplication()public
34    {
35        ans=val1*val2;
36    }
37
38    function division()public
39
40

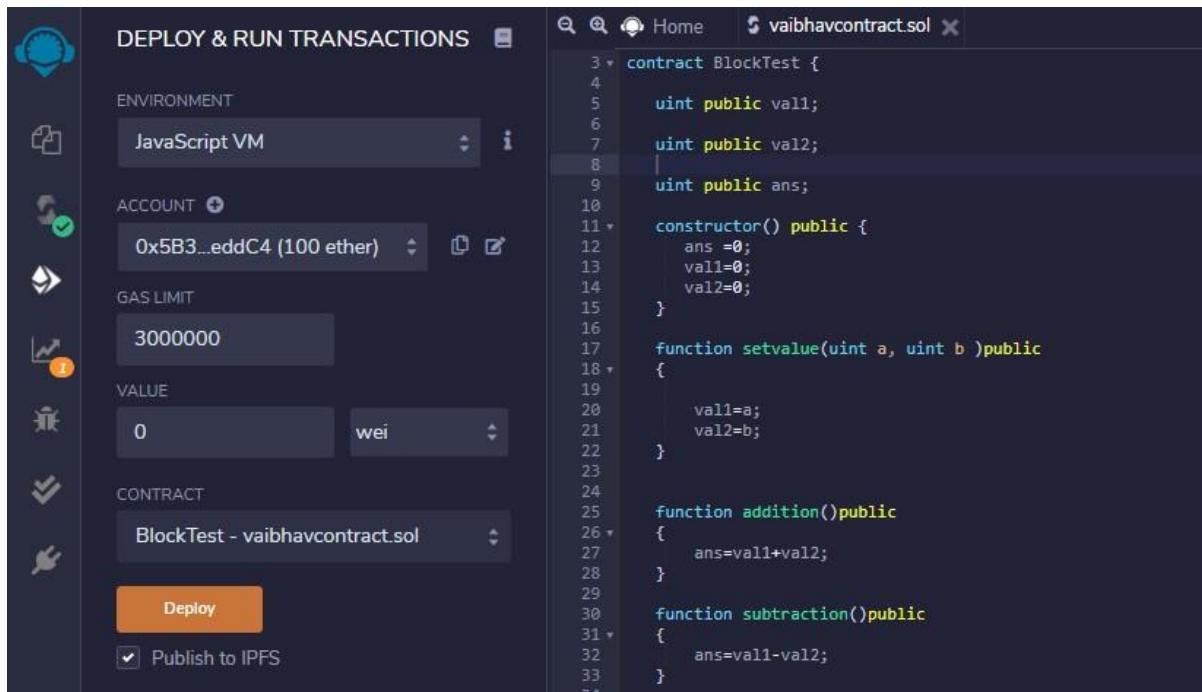
```

Step 4:- After all code in the editor will compile the first application in remix IDE. before compiling will have to set the compiler and EVM version.



The screenshot shows the Solidity Compiler interface within the Remix IDE. The left sidebar includes settings for the COMPILER (version 0.5.17+commit.d19bba13), LANGUAGE (Solidity), EVM VERSION (compiler default), and COMPILER CONFIGURATION (Auto compile, Enable optimization set to 200, Hide warnings). The right side shows the Solidity code for the 'BlockTest' contract, identical to the one in the previous screenshot. A prominent blue button at the bottom left says 'Compile vaibhavcontract.sol'. The CONTRACT dropdown at the bottom also lists 'BlockTest (vaibhavcontract.sol)'.

Step 5: after successful compilation code deploy the file for which we have to set the environment as well as the any account for transaction and gas limit.



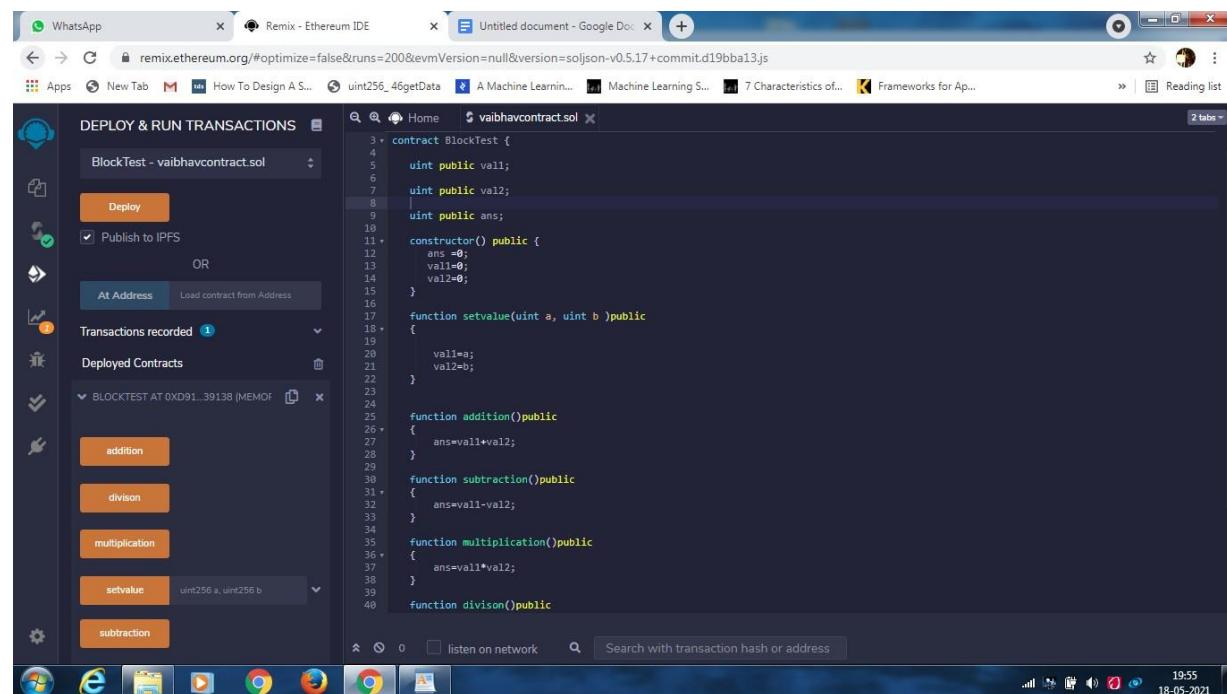
The screenshot shows the Remix Ethereum IDE interface. On the left, there's a sidebar with various icons for deploying, running, and managing contracts. The main area is titled "DEPLOY & RUN TRANSACTIONS". It includes sections for "ENVIRONMENT" (set to "JavaScript VM"), "ACCOUNT" (showing an account with address 0x5B3...eddC4 and 100 ether balance), "GAS LIMIT" (set to 3000000), "VALUE" (set to 0 wei), and "CONTRACT" (selected as "BlockTest - vaibhavcontract.sol"). Below these are buttons for "Deploy" and "Publish to IPFS". To the right, the code editor displays the Solidity source code for "vaibhavcontract.sol".

```

3+ contract BlockTest {
4+     uint public val1;
5+     uint public val2;
6+     uint public ans;
7+
8+     constructor() public {
9+         ans =0;
10+        val1=0;
11+        val2=0;
12+    }
13+
14+    function setvalue(uint a, uint b )public
15+    {
16+        val1=a;
17+        val2=b;
18+    }
19+
20+    function addition()public
21+    {
22+        ans=val1+val2;
23+    }
24+
25+    function subtraction()public
26+    {
27+        ans=val1-val2;
28+    }
29+
30+    function multiplication()public
31+    {
32+        ans=val1*val2;
33+    }
34+
35+    function division()public
36+    {
37+        ans=val1/val2;
38+    }
39+
40+}

```

Step 6: after deployment go to deployed contract and execute the functions one by one for calculations.



This screenshot shows the Remix Ethereum IDE after the contract has been deployed. The "Deployed Contracts" section now lists the deployed contract "BLOCKTEST AT 0xD91...39138 (MEMO)". Below it, a list of functions is provided: "addition", "division", "multiplication", "setvalue", and "subtraction". The "setvalue" function is currently selected, showing its parameters: "uint256 a, uint256 b". The rest of the interface remains similar to the deployment screen, with the code editor still showing the same Solidity code.

Code:

```
pragma solidity ^0.5.0;
contract BlockTest {
    uint public val1;
    uint public val2;
    uint public ans;
    constructor()
    public {

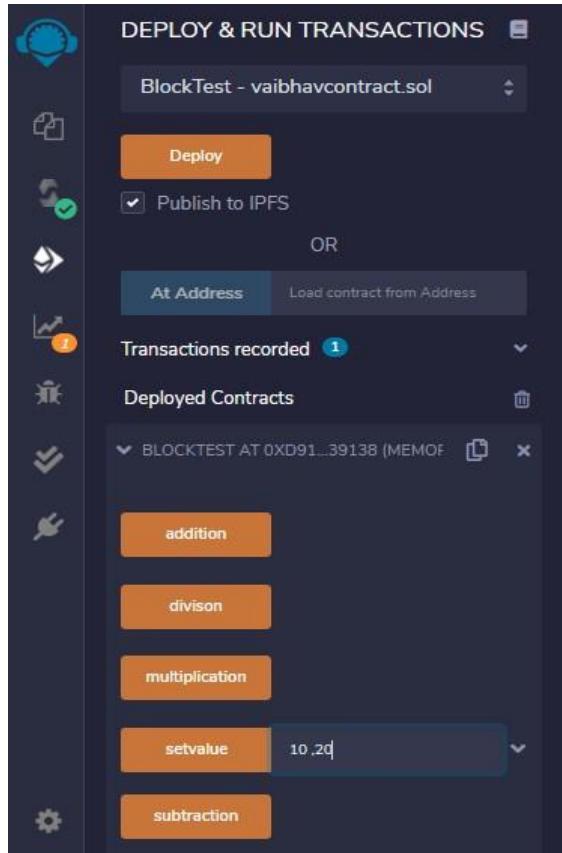
        ans=0;val1=
        0;v al2=0;
    }
    function setvalue(uint a, uint b )public

    {
        val1=a;
        val2=b;
    }
    function addition()public
    {
        ans=val1+val2;
    }
    function subtraction()public
    {
        ans=val1-val2;
    }
    function multiplication()public
    {
        ans=val1*val2;
    }
}
```

```
function divison()public
{
    ans=val1/val2;
}
//public function
function getData() public view returns(uint)
{ return ans; }
```

Output:-

Set the value as 10 and 20



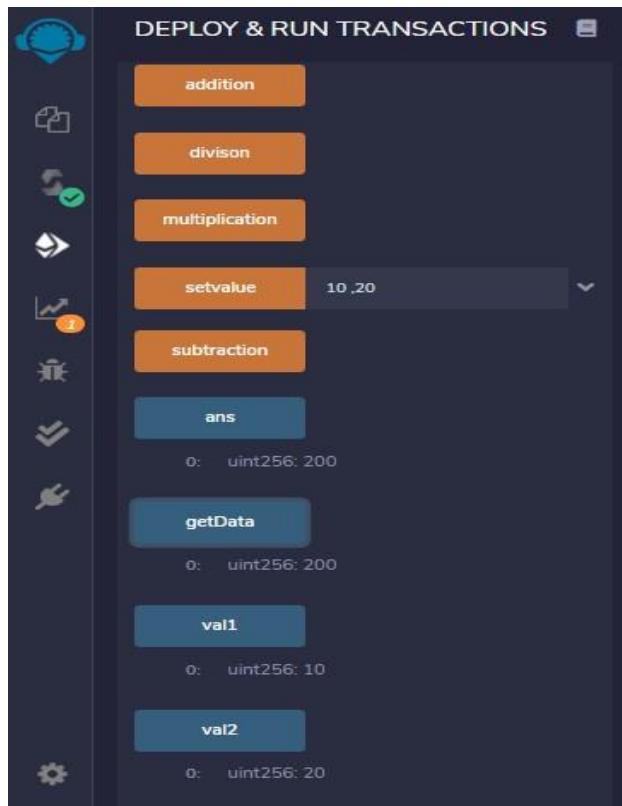
After setting the value of two state variables we perform the addition as a operation.

The screenshot shows the Truffle UI interface with the Solidity code for the contract "vaibhavcontract.sol" displayed on the right. The code defines four functions: addition, subtraction, multiplication, and division, each taking two uint256 parameters and returning a uint256 result. It also includes a public function getData that returns the value of the state variable ans. On the left, there is a sidebar with buttons for "addition", "divison", "multiplication", "setvalue" (with a value of "10.20"), "subtraction", "ans", "getData", "val1", and "val2". Below these buttons, their current values are listed: "ans" is 0, "getData" is 0, "val1" is 0, and "val2" is 0. The bottom of the screen shows a transaction history with one entry: "[call] from: 0x58380a6a701c568545dCfcB03FcB875f56beddC4 to: BlockTest.val2() data: 0x95c...acbe0". There is also a "Debug" button at the bottom right.

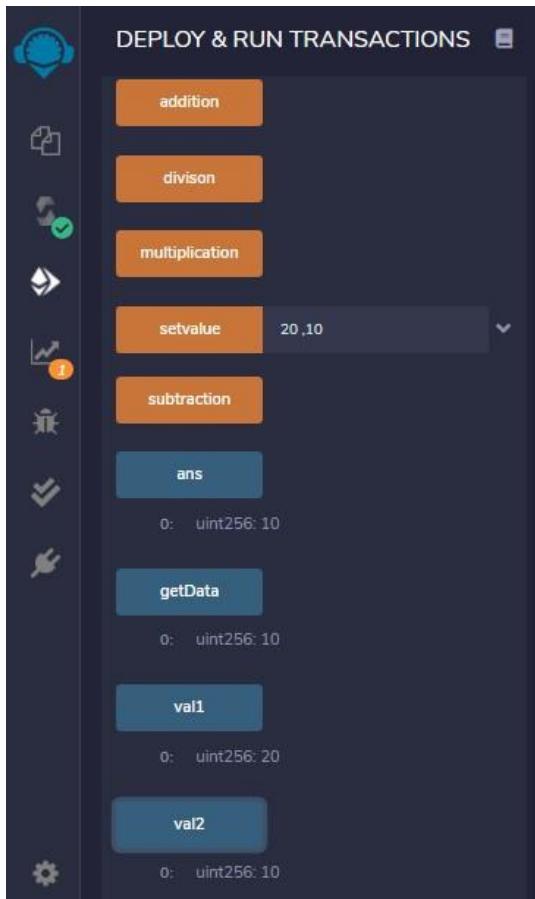
```

19 val1=a;
20 val2=b;
21 }
22
23
24
25 function addition()public
26 {
27     ans=val1+val2;
28 }
29
30
31 function subtraction()public
32 {
33     ans=val1-val2;
34 }
35
36 function multiplication()public
37 {
38     ans=val1*val2;
39 }
40
41 function division()public
42 {
43     ans=val1/val2;
44 }
45
46 //public function
47
48 function getData() public view returns(uint) { return ans; }
49
50
51

```



After multiplication we got 200 as a answer



After subtraction we get 10 (20,10)

ii) Implement constructor .

Steps:

Step 1: Create a contract in remix ethereum IDE using solidity as a programming language name the contract and create one constructor using modifier as a public.

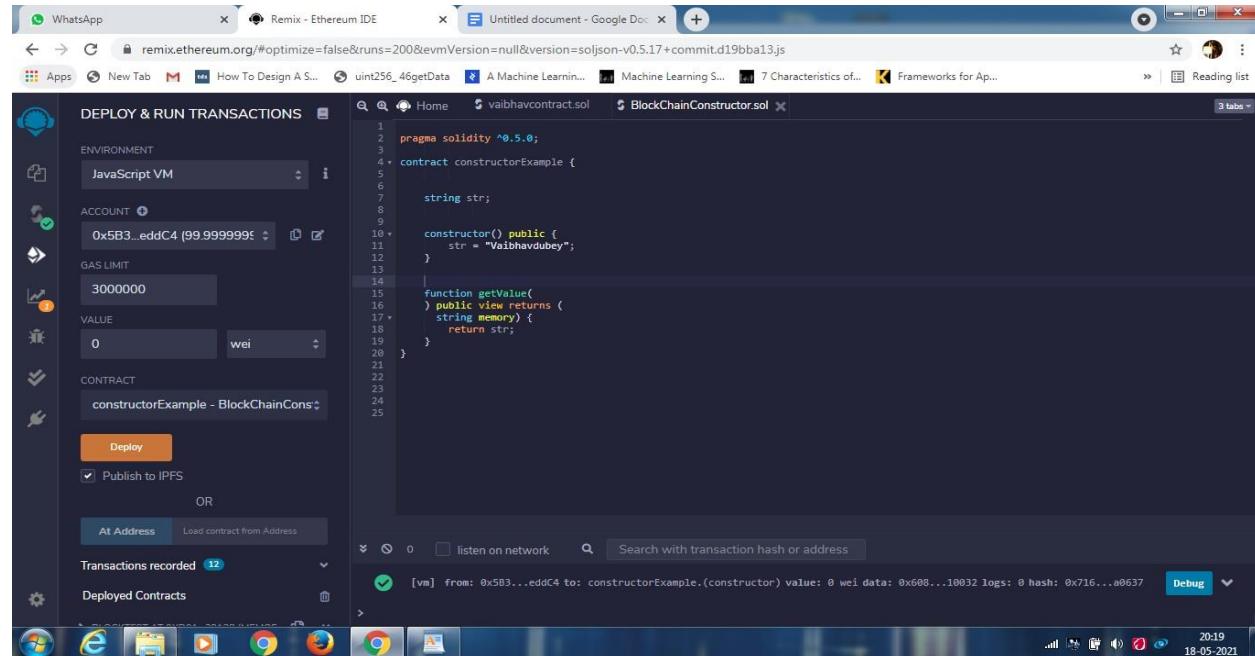
```

1 pragma solidity ^0.5.0;
2
3 contract constructorExample {
4
5     string str;
6
7     constructor() public {
8         str = "Vaibhavdubey";
9     }
10
11     function getValue()
12         public view returns (
13             string memory) {
14         return str;
15     }
16
17 }
18
19
20
21
22
23
24
25

```

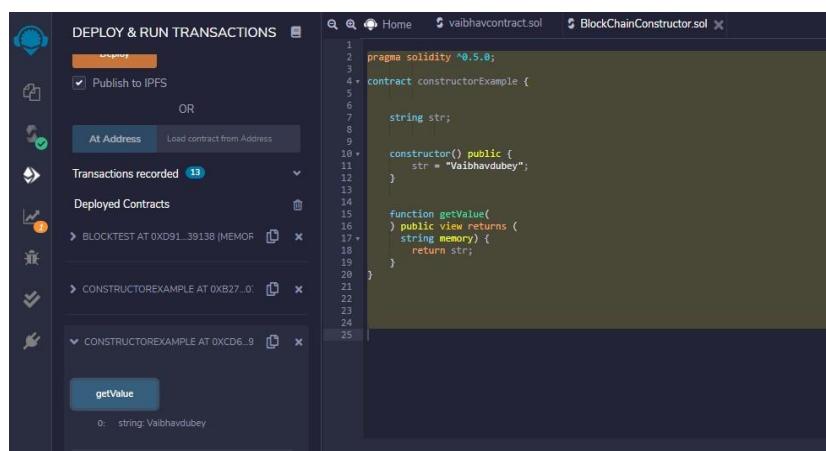
Step 2: After writing the code in the editor compile the Application And for compilation we have to set compiler gas value.

Step 3: after the compilation of the application it is ready to deploy and run. We kept the gas value and the environment default.



```
pragma solidity ^0.5.0;
contract constructorExample {
    string str;
    constructor() public{
        str = "Vaibhavdubey";
    }
    function getValue()
        public view returns (
            string memory) {
        return str;
    }
}
```

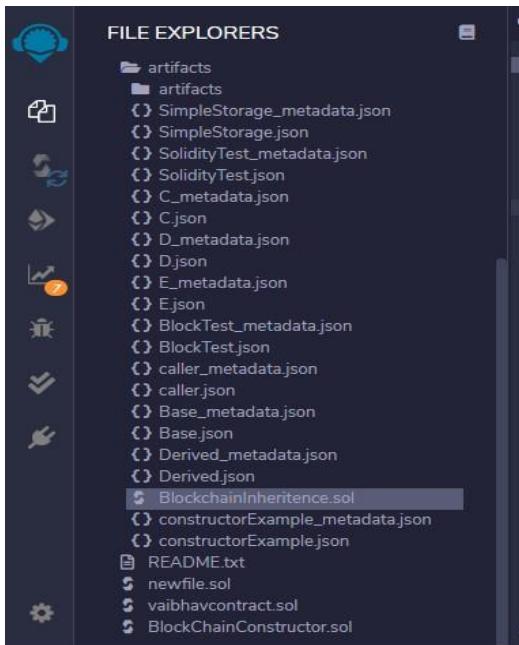
We got the output as a string using constructor in public scope.



ii) Implement Inheritance Steps:

Step 1: Create one Blockchain application for inheritance by creating one solidity file in remix etherum IDE.

Single level inheritance



Step 2: Create one base contract and one child contract and test the inheritance by appropriate function by the caller contract.

```

function setValue() external {
    uint a = 50;
    uint b = 20;
    sum = a + b;
}

// Defining child contract
contract child is parent{
    function getValue()
        external view returns(uint) {
            return sum;
    }
}

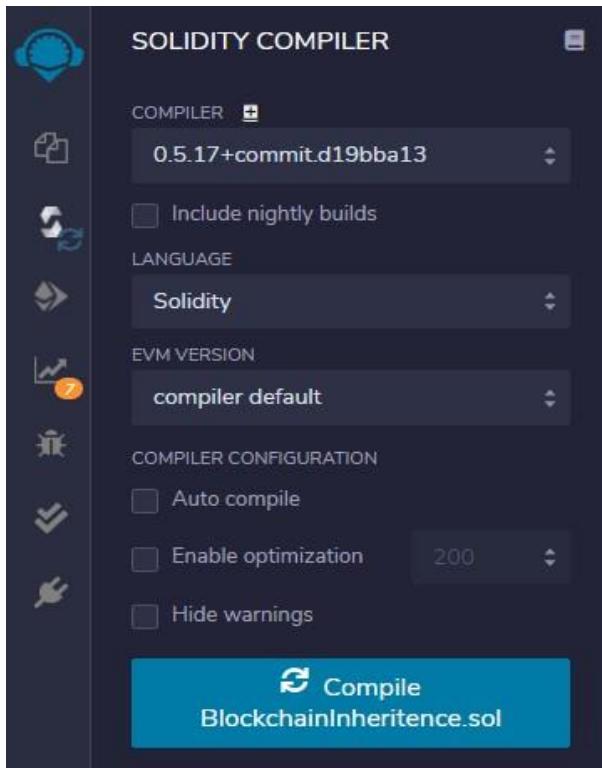
contract caller {
    child cc = new child();

    function testInheritance()
        public returns (uint) {
            cc.setValue();
            return cc.getValue();
    }
}

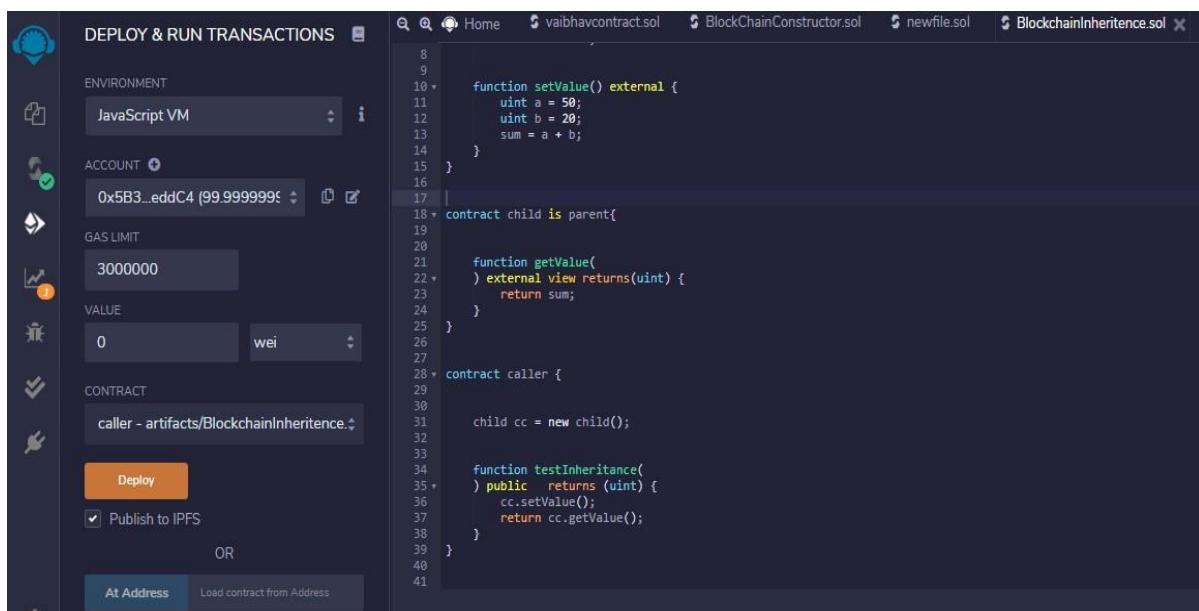
```

The screenshot shows the Solidity code for a contract named 'BlockchainInheritance'. It includes a base contract 'parent' with a function 'setValue' that adds two uint variables 'a' and 'b' and stores the result in 'sum'. It also defines a child contract 'child' that inherits from 'parent'. The child contract has a function 'getValue' that returns the value of 'sum'. Finally, it defines a 'caller' contract that creates an instance of 'child' and calls its 'testInheritance' function, which in turn calls 'setValue' on the child contract and returns its value.

Step 3: Compile the Blockchain application after the coding in the editor panel.



Step 4: After successful compilation of the code deploy the application for execution.

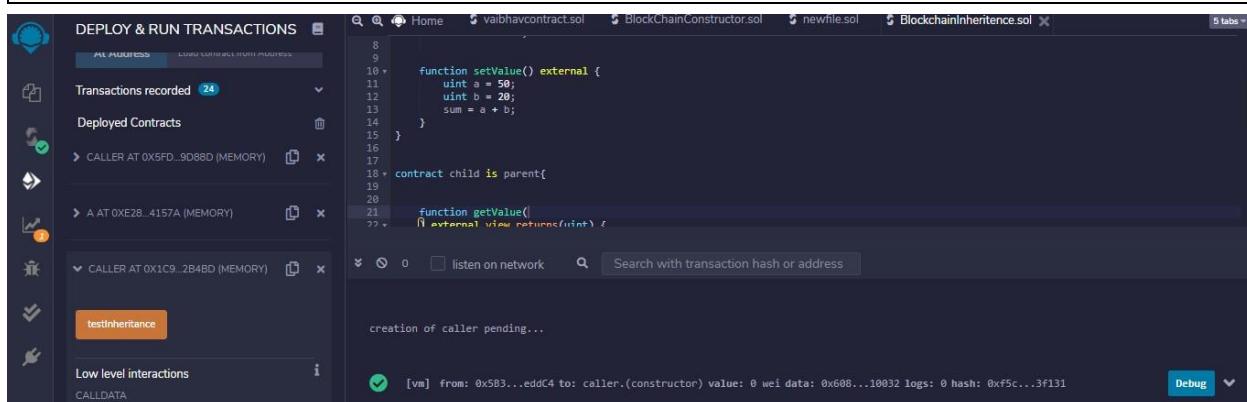


Output:

After the deployment of the application Run the application from deployed contracts .

Code

```
pragma solidity >=0.4.22 <0.6.0;
contract parent{
    uint internal sum;
    function setValue() external {
        uint a = 50;
        uint b = 20;
        sum = a + b;
    }
} contract child is parent{
    function getValue()
    external view returns(uint) {
        return sum;
    }
}
contract caller {
    child cc = new child();
    function testInheritance()
    public returns (uint) {
        cc.setValue();
        return cc.getValue();
    }
}
```



After clicking on `testInheritance` from deployed contracts we got the result by clicking on the green right symbol.

```

10+     function setValue() external {
11+         uint a = 50;
12+         uint b = 20;
13+         sum = a + b;
14+     }
15+
16+
17+ contract child is parent{
18+
19+
20+     function getValue()
21+     external view returns(uint) {
22+         return sum;
23+     }
24+
25+     function sum() external view returns(uint) {
26+         return sum;
27+     }
28+ }
29+
30+ }
```

Transactions recorded: 24 Deployed Contracts: A AT 0XE28..4157A (MEMORY)

testInheritance

Low level interactions: CALldata

Gas: 3000000 gas, Transaction cost: 47455 gas, Execution cost: 26183 gas, Hash: 0x4a7d381ce8f032e0009e6efa8fe15a8c8feeb820ae17657bia2b439f65ff72, Input: 0xd93...74dd5, Decoded input: {}, Decoded output: { "a": "uint256: 70" }

Result is 70.

iv) Implement the interface in blockchain.

Steps :

Step 1:- Create one solidity file in Remix etheruem IDE , by creating one interface in the base contract and create one derived contract .

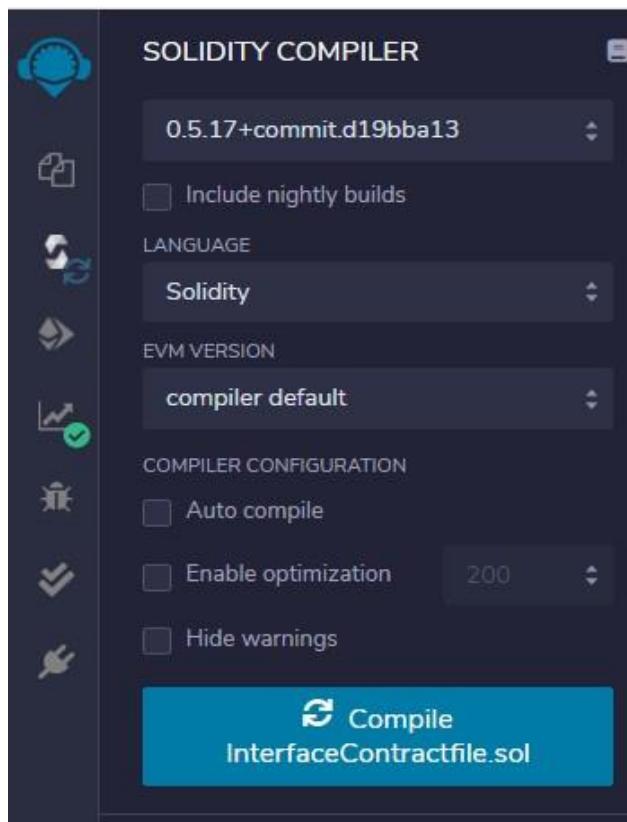
```

1 pragma solidity ^0.5.0;
2
3 interface Sample {
4     function answer() external view returns(uint);
5 }
6 contract Check is Sample {
7     constructor() public {}
8     function answer() external view returns(uint){
9         uint a = 10;
10        uint b = 52;
11        uint result = a * b;
12        return result;
13    }
14 }
```

FILE EXPLORERS: SolidityTest.json, C_metadata.json, C.json, D_metadata.json, D.json, E_metadata.json, E.json, BlockTest_metadata.json, BlockTest.json, caller_metadata.json, caller.json, Base_metadata.json, Base.json, Derived_metadata.json, Derived.json, constructorExample_metadata.json, constructorExample.json, BlockchainInheritance.sol, Calculator_metadata.json, Calculator.json, Test_metadata.json, Test.json, InterfaceContractfile.sol, README.txt, vaibhavcontract.sol, BlockChainConstructor.sol, newfile.sol

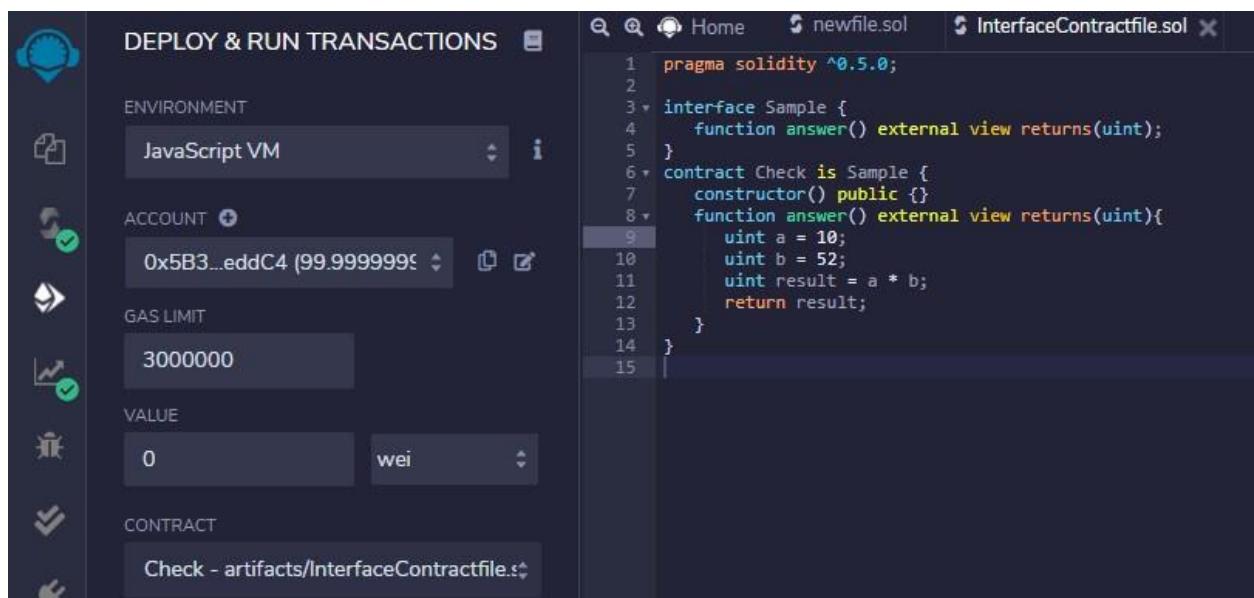
ContractDefinition Calculator → 1 reference(s)

Step 2: After writing the code in te editor implement the interface by the derived contract Complie the Application .

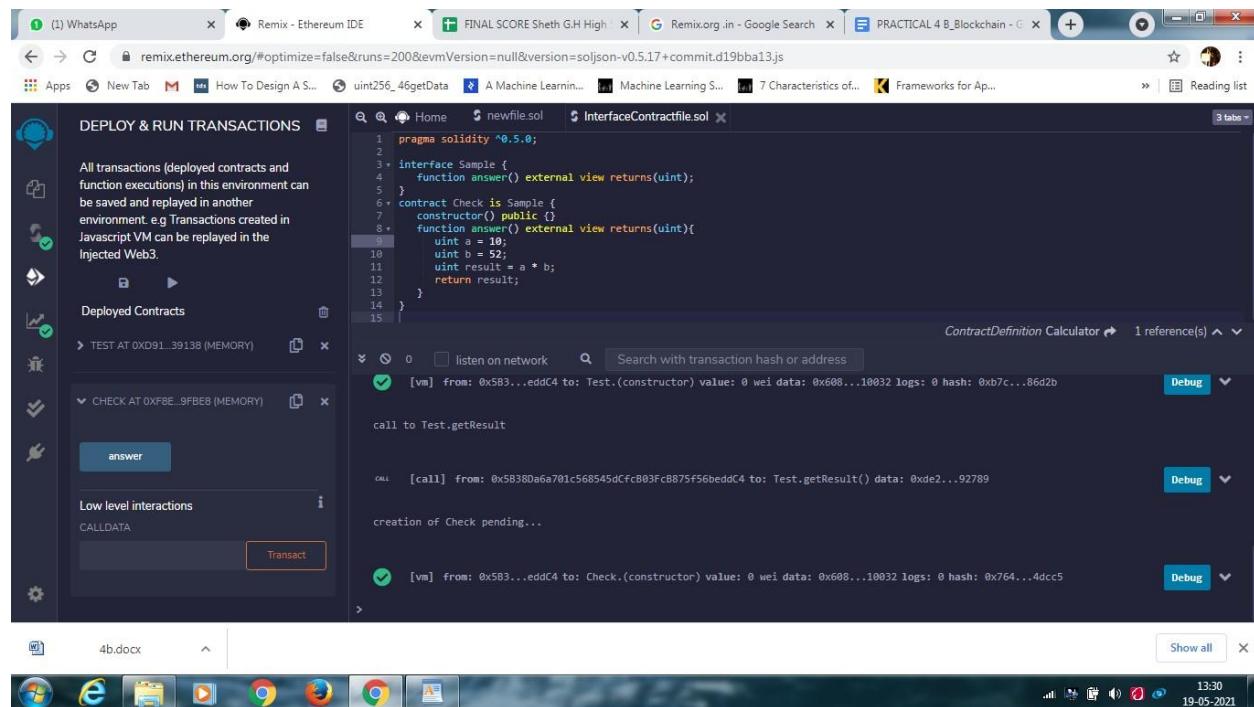


Step 3: After the successful compilation deploy the blockchain application for both the Contracts (Base and Derived)

For which we have to select the contract from dropdown box.



Step 4: after the deployment now run or execute the application from deployed contracts.



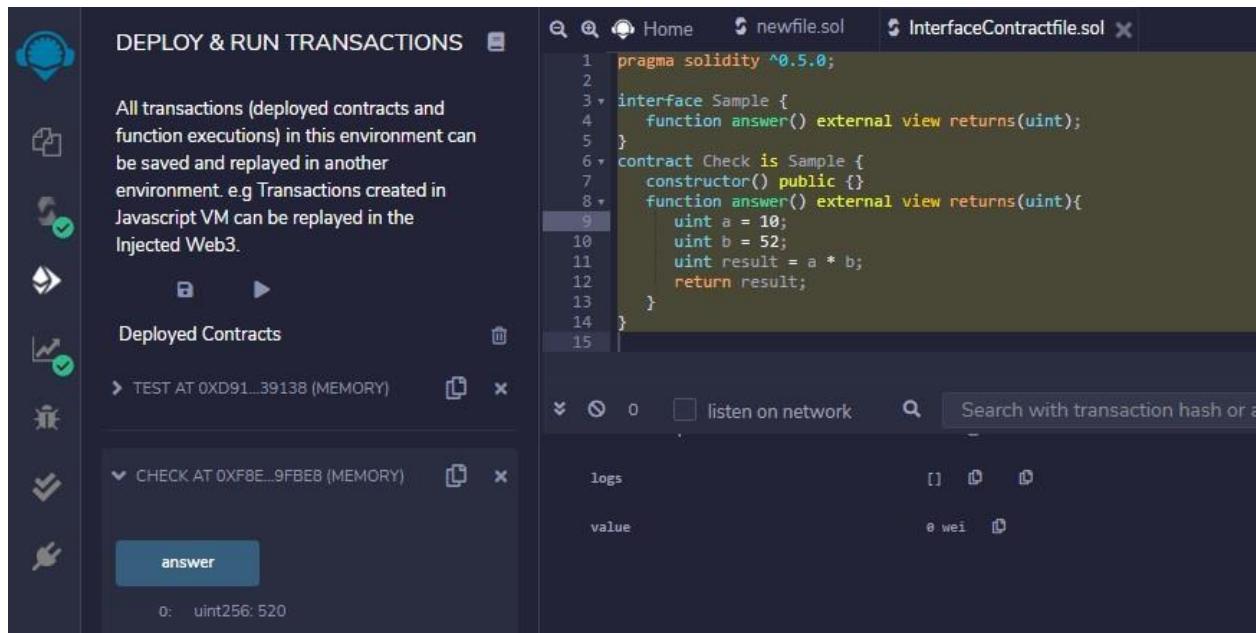
Code

```
pragma solidity ^0.5.0;

interface Sample {
    function answer() external view returns(uint);
} contract Check is Sample {    constructor() public {}
    function answer() external view returns(uint)
{
    uint a = 10; uint b = 52;
    uint result = a * b;
    return result;
}
```

Output:

In the deployed contracts we select the contract in simply click on the tab of answer and got the output.



```

1 pragma solidity ^0.5.0;
2
3 interface Sample {
4     function answer() external view returns(uint);
5 }
6 contract Check is Sample {
7     constructor() public {}
8     function answer() external view returns(uint){
9         uint a = 10;
10        uint b = 52;
11        uint result = a * b;
12        return result;
13    }
14 }
15

```

The Remix IDE interface shows the following details:

- Deploy & Run Transactions:** A sidebar with various icons for deployment, running, and monitoring.
- Deployed Contracts:** A list of contracts deployed in memory, including "TEST AT 0xD91...39138 (MEMORY)" and "CHECK AT 0xF8E...9FBEB (MEMORY)".
- Logs:** An empty logs section.
- Value:** A value of 0 wei.
- Search:** A search bar for transaction hash or address.

Answer is 520

v) Implement the Abstract contract.

Steps:

Step 1:

Create the solidity file in the Remix IDE and used the abstract keyword to create the abstract contract .



Step 2: Write the code in the editor and compile the code through default compiler.

```

pragma solidity ^0.5.0;
contract Base {
    function getResult() public view returns(uint);
}
contract Adder is Base {
    function getResult() public view returns(uint) {
        uint a = 19;
        uint b = 20;
        uint result = a + b;
        return result;
    }
}

```

Step 3: After successful implementation deploy the code.

But before deploying select the derived contract from the contract dropdown box.

DEPLOY & RUN TRANSACTIONS

VALUE: 0 wei

CONTRACT: Adder - artifacts/artifacts/abstractinter...

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 4

All transactions (deployed contracts and function executions) in this environment can be saved and replayed in another environment. e.g. Transactions created in Javascript VM can be replayed in the Injected Web3.

creation of Adder pending...

[vm] from: 0x583...eddC4 to: Adder.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0x601...6c2dd

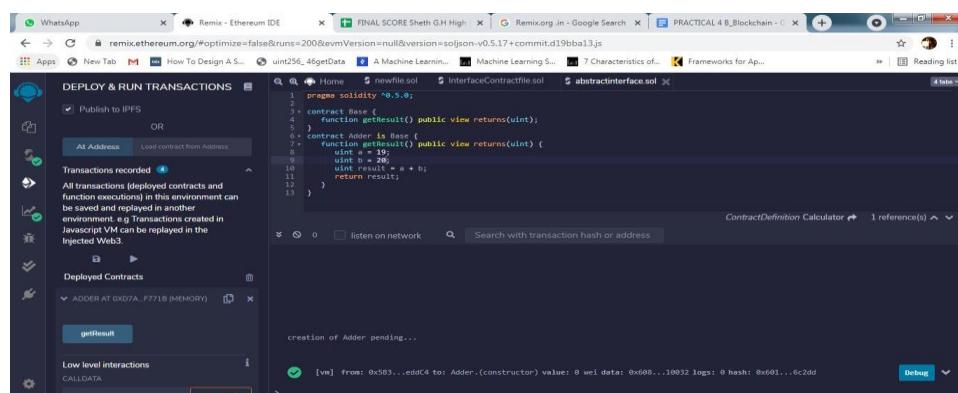
Step 4: After the deployment of the application execute (run) the function (tab) from deployed contract.

Code

```
pragma solidity ^0.5.0;

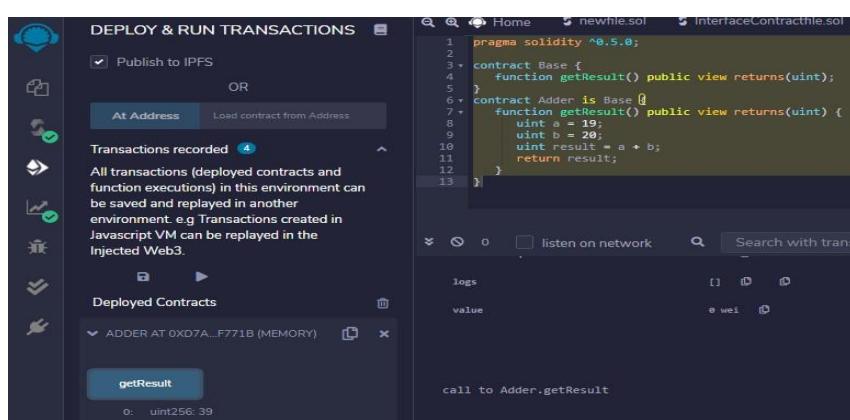
contract Base {
    function getResult() public view returns(uint);
}

contract Adder is Base {
    function getResult() public view returns(uint) {
        uint a = 19; uint b = 20;
        uint result = a + b;
        return result;
    }
}
```



Output:

After the deployment of the code click on the getResult tab as function and record the result.



PRACTICAL 4C:

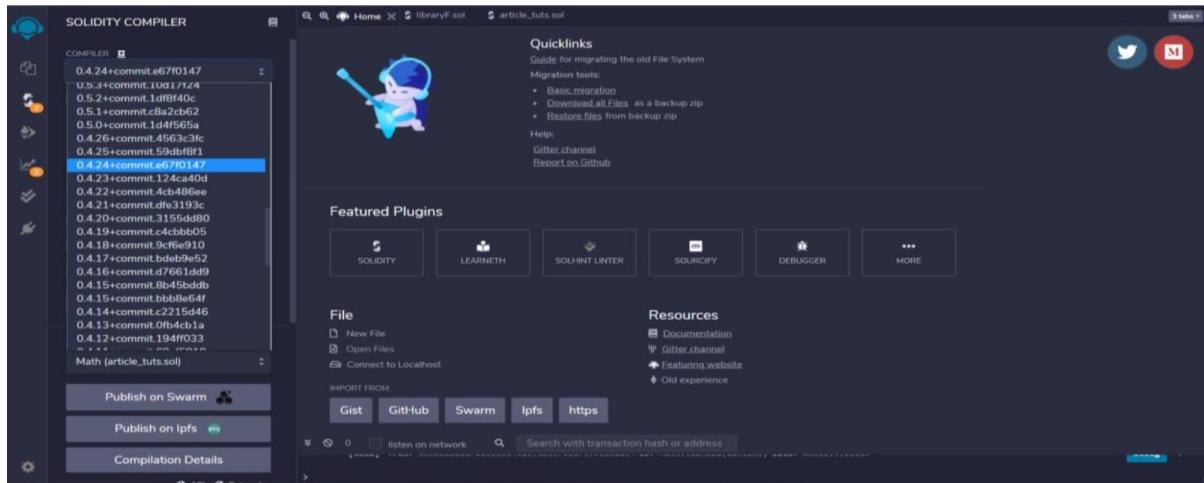
Aim:

Implement and demonstrate the use of the following in Solidity
Libraries, Assembly, Events, Error handling.

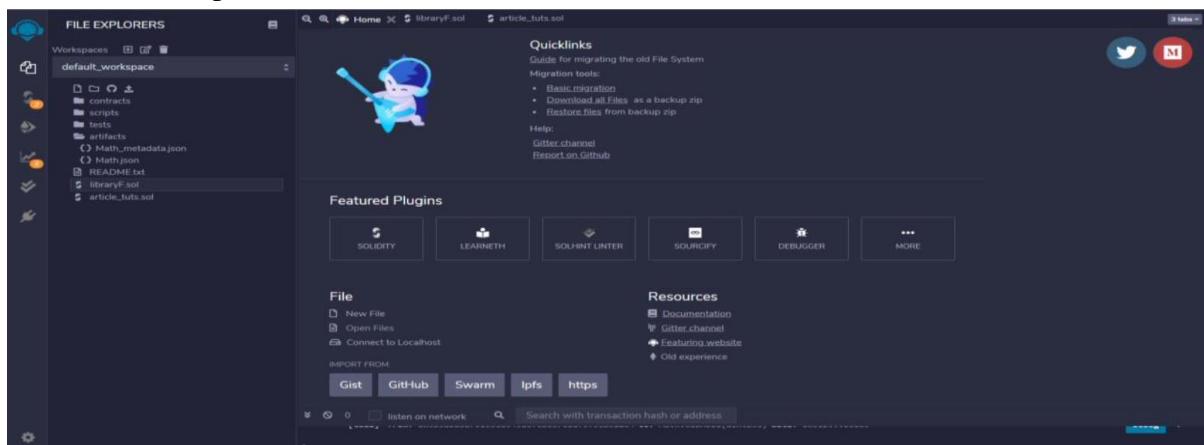
Libraries: To Implement and Demonstrate the use of Solidity Libraries

STEPS:

1. Open Remix on your browser and do the following settings;

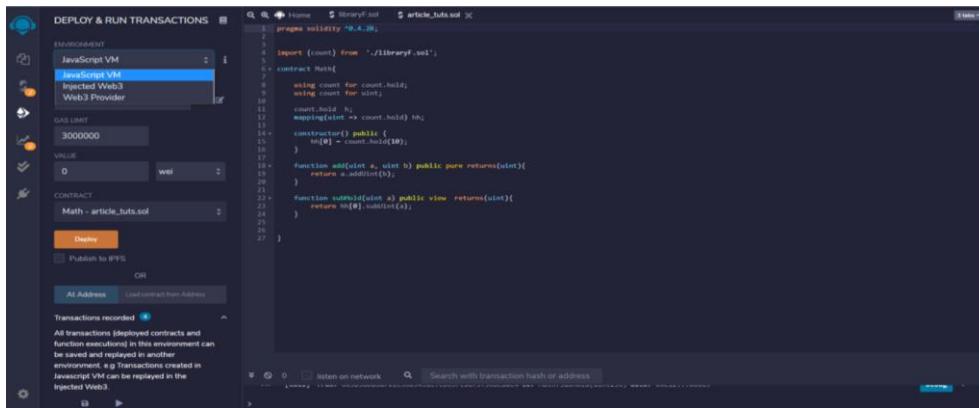


2: Choose a compiler (In this case version:0.4.24)

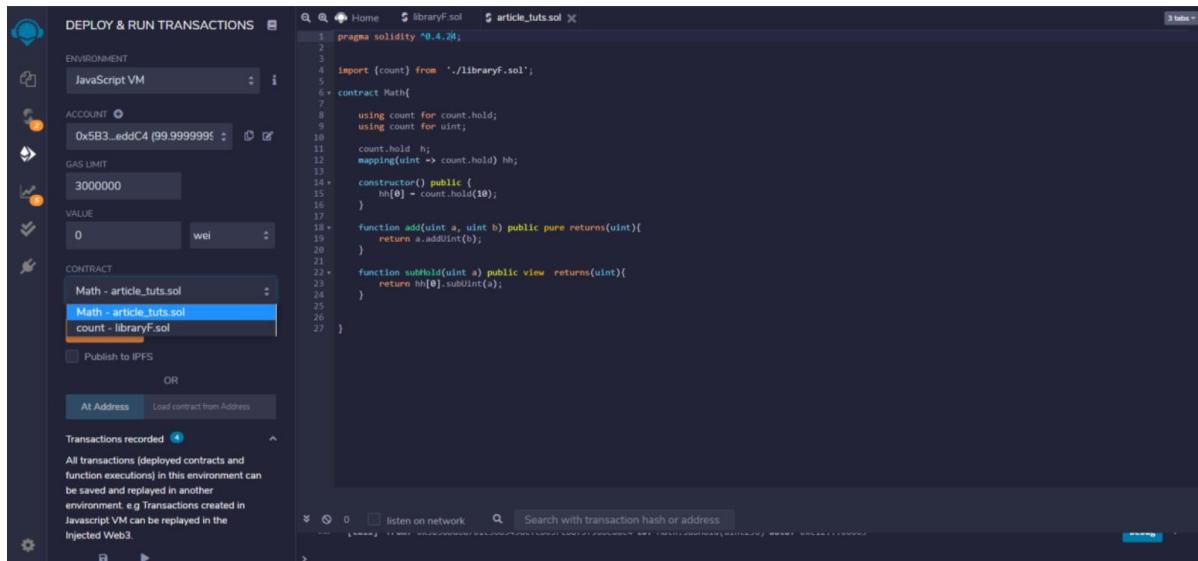


Write the code in the library.sol and article_tuts.sol.

To deploy the code, choose the environment JavaScript VM



6. Deploy the contract Math



7. Perform the various operations (add and subHold)

CODE :

```
pragma solidity ^0.4.24;
library count{
    struct hold{
        uint a;
    }
    function subUint(hold storage s, uint b) public returns(uint){
        require(s.a >= b); // Make sure it doesn't return a negative value.
        return s.a - b;
    }
    function addUint(uint a , uint b) public pure returns(uint)
        uint c = a + b;
        require(c >= a); // Make sure the right computation was made return c;
    }
```

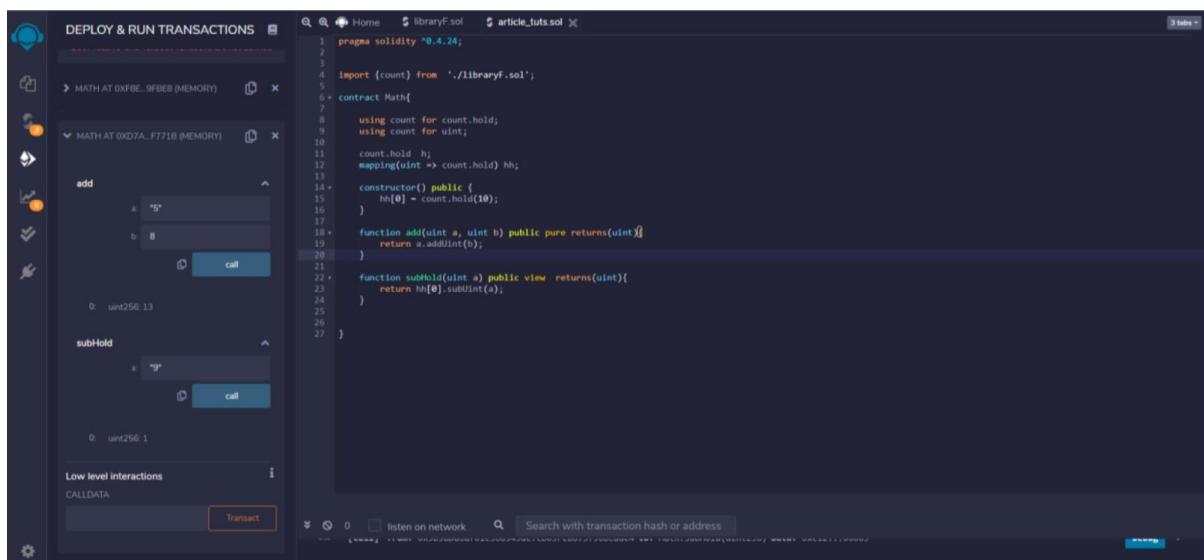
```

}

pragma solidity ^0.4.24;
import {count} from './libraryF.sol';
contract Math{
    using count for count.hold; using count for uint;
    count.hold hh;
    mapping(uint => count.hold) hh;
    constructor() public {
        hh[0] = count.hold(10);
    }
    function add(uint a, uint b) public pure returns(uint){
        return a.addUint(b);
    }
    function subHold(uint a) public view returns(uint){
        return hh[0].subUint(a);
    }
}

```

Output:



Assembly: To Implement and Demonstrate the use of Solidity Assembly.

STEPS:

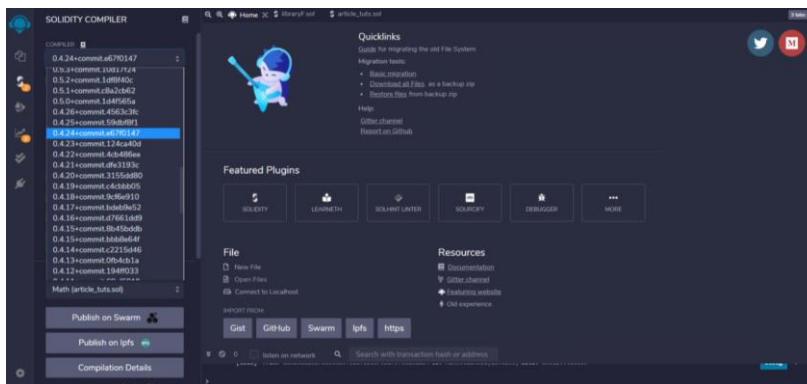
Inline Assembly: Inline assembly code can be written inside solidity code for more fine-grain control and especially used for enhancing the language via creating new libraries. Inline assembly can be inserted in between solidity statements in a way that EVM can understand. It can also be used when the optimizer is not able to produce efficient code. Solidity becomes easier when features like assembly local variables, functions calls, switch statements, if statements, loops, etc. are used.

Syntax

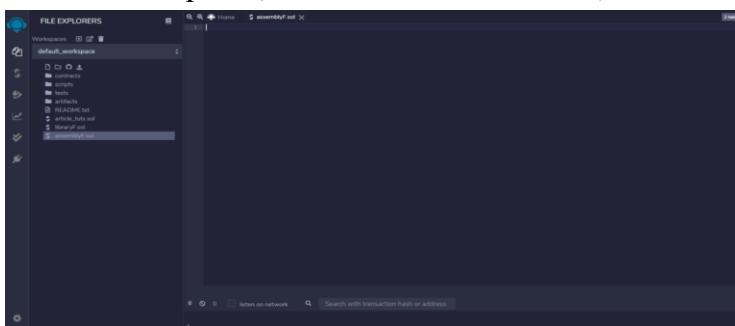
```
assembly {
```

```
// assembly language statements
}

Open Remix on your browser and do the following settin
```

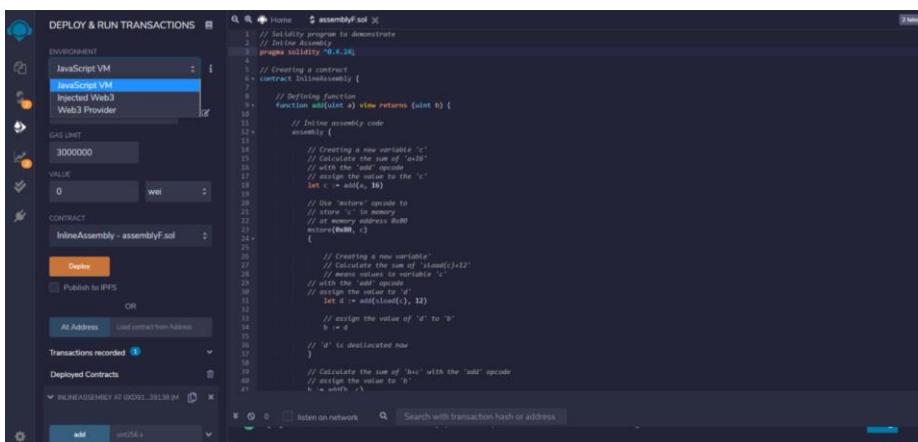


Choose a compiler (In this case version:0.4.24)

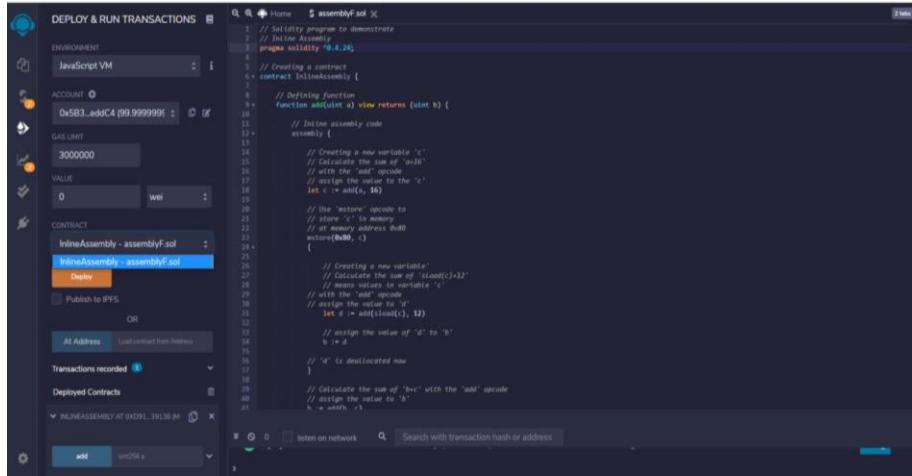


Write the code in the assemblyF.sol.

To deploy the code, choose the environment JavaScript VM.



Deploy the contract InnerAssembly – assembly.sol.



Perform the add operation.

CODE

```
// Solidity program to demonstrate // Inline Assembly
pragma solidity ^0.4.24;

// Creating a contract
contract InlineAssembly { // Defining function
    function add(uint a) view returns (uint b) { // Inline assembly code
        assembly {
            // Creating a new variable 'c'
            // Calculate the sum of 'a+16' // with the 'add' opcode
            // assign the value to the 'c' let c := add(a, 16)
            // Use 'mstore' opcode to // store 'c' in memory
            // at memory address 0x80 mstore(0x80, c)
            // Creating a new variable'
            // Calculate the sum of 'sload(c)+12' // means values in variable 'c'
            // with the 'add' opcode
            // assign the value to 'd'
            let d := add(sload(c), 12)
            // assign the value of 'd' to 'b' b := d
            // 'd' is deallocated now }
            // Calculate the sum of 'b+c' with the 'add' opcode
            // assign the value to 'b'
            b := add(b, c)
            // 'c' is deallocated here }
        }
    }
}
```

OUTPUT:

```
// Solidity program to demonstrate
// InLine Assembly
pragma solidity ^0.4.24;

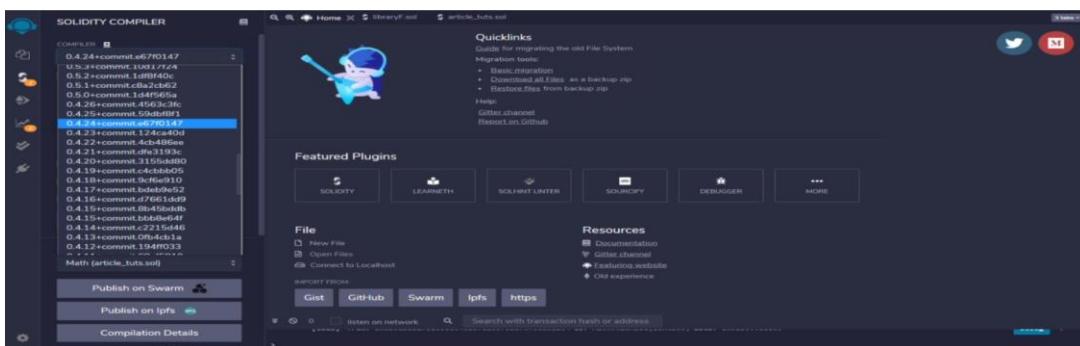
contract InLineAssembly {
    // Defining function
    function add(uint a) view returns (uint b) {
        // Inline assembly code
        assembly {
            // Creating a new variable 'c'
            // with the "mstore" opcode
            // at memory address 0x00
            c := add(a, 10)
            // Use "swtore" opcode to
            // store the value in memory
            // at memory address 0x00
            sstore(0x00, c)
        }
        // Creating a new variable
        // Calculate the sum of "load(c)+10"
        // means values in variable 'c'
        // plus 10
        // assign the value to 'd'
        let d := add(c, 10)
        // assign the value of 'd' to 'b'
        b := d
        // 'd' is deallocated now
    }
    // Calculate the sum of 'b+c' with the "add" opcode
    // assign the value to 'b'
    b := add(b, c)
}

```

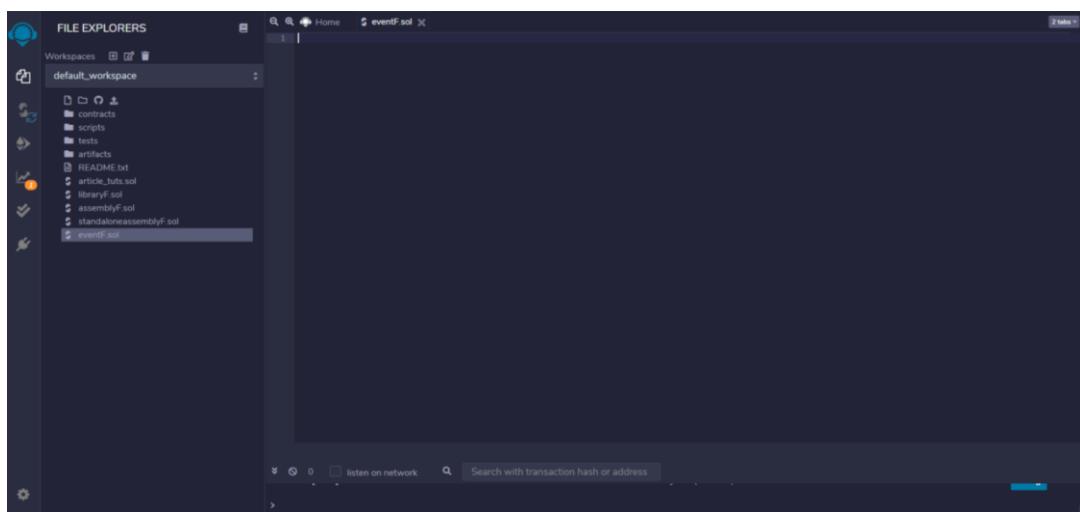
Events: To Implement and Demonstrate the use of Solidity Events

STEPS:

Open Remix on your browser and do the following settings;

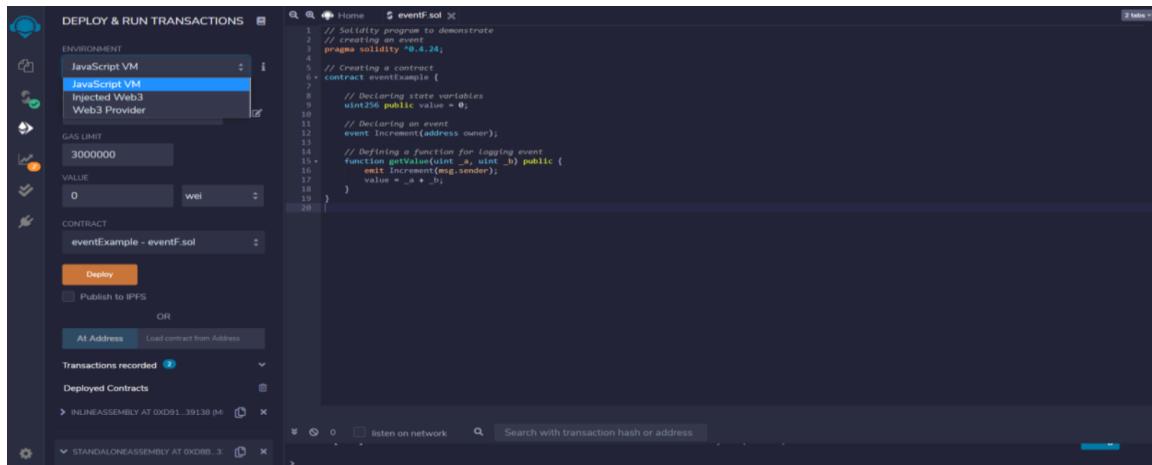


Choose a compiler (In this case version:0.4.24)

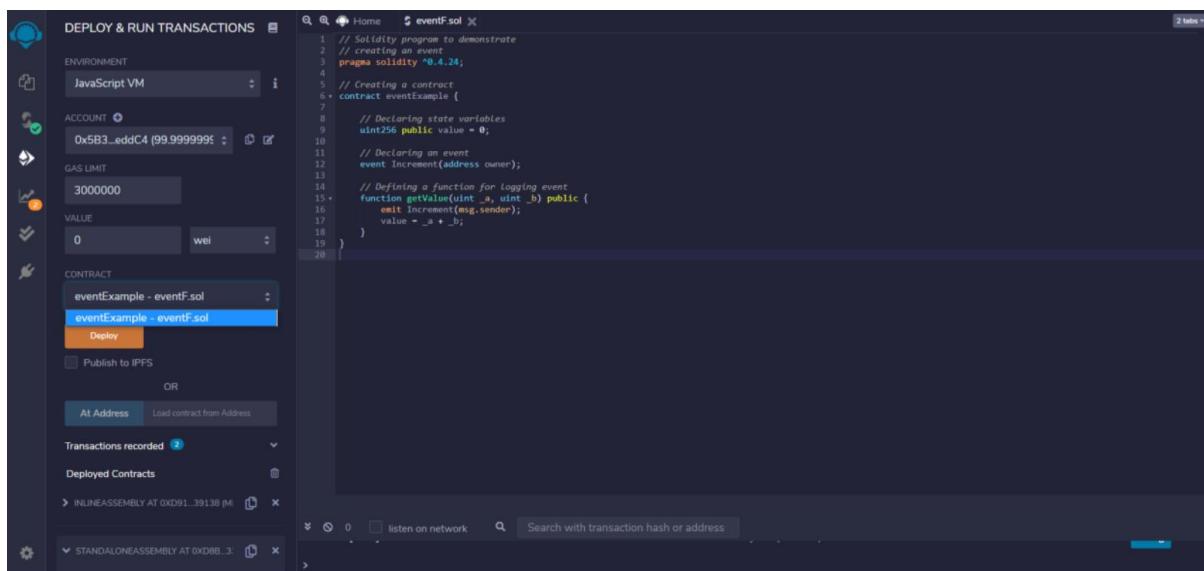


Write the code in the eventF.sol.

To deploy the code, choose the environment JavaScript VM.



Deploy the contract eventExample-eventF.sol.



Perform the getValue

CODE:

```
// Solidity program to demonstrate // creating an event
pragma solidity ^0.4.24;
// Creating a contract
contract eventExample {
// Declaring state variables
uint256 public value = 0;
// Declaring an event
event Increment(address owner);
// Defining a function for logging event
function getValue(uint _a, uint _b) public {
emit Increment(msg.sender);
value = _a + _b;
}
```

OUTPUT

```

Solidity version: 0.4.24
eventExample.sol:1
1 // SPDX-License-Identifier: MIT
2 // Creating an event
3 // Version 0.4.24
4 // Creating a contract
5 contract eventExample {
6     // Declaring state variables
7     uint256 public value;
8
9     // Defining an event
10    event increment(address owner);
11
12    // Defining a function for logging event
13    function increment(uint256 _val) public {
14        emit increment(msg.sender);
15        value = _val;
16    }
17}

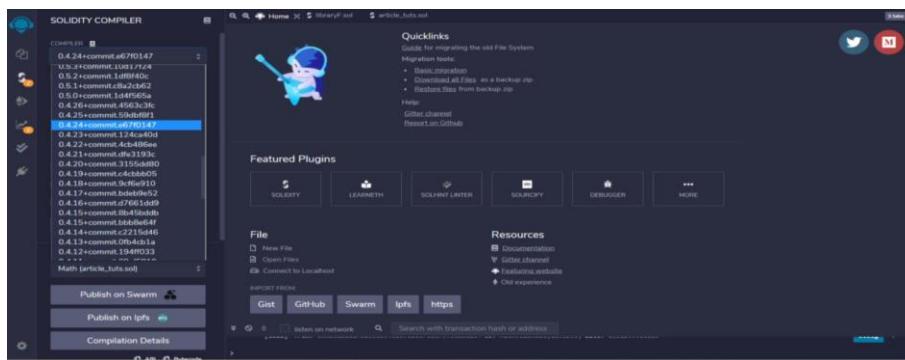
```

The interface shows the contract deployed at address 0x605... and the event example deployed at address 0x605... . The 'getValue' function is called with the value 10.

Error Handling: To Implement and Demonstrate the use of Solidity Error Handling

STEPS :Solidity has many functions for error handling. Errors can occur at compile time or runtime. Solidity is compiled to byte code and there a syntax error check happens at compile-time, while runtime errors are difficult to catch and occurs mainly while executing the contracts. Some of the runtime errors are out-of-gas error, data type overflow error, divide by zero error, array-out-of-index error, etc. Until version 4.10 a single throw statement was there in solidity to handle errors, so to handle errors multiple if...else statements, one has to implement for checking the values and throw errors which consume more gas. After version 4.10 new error handling construct assert, require, revert statements were introduced and the throw was made absolute.

1. Open Remix on your browser and do the following settings;



2. Choose a compiler (In this case version:0.4.24)

```

// Solidity program to demonstrate require
// statement
pragma solidity ^0.4.24;

// Creating a contract
contract requirement {
    // Defining function to check input
    function eventExample(uint _input) public view returns(string memory){
        string memory;
        require(_input >= 0, "invalid uint8");
        require(_input <= 255, "invalid uint8");
        return "Input is UInt8";
    }
    // Defining function to use require statement
    function Odd(uint _input) public view returns(bool){
        require(_input % 2 != 0);
        return true;
    }
}

```

Write the code in the requireF.sol.

To deploy the code, choose the environment JavaScript VM.

```

// Solidity program to demonstrate require
// statement
pragma solidity ^0.4.24;

// Creating a contract
contract requirement {
    // Defining function to check input
    function eventExample(uint _input) public view returns(string memory){
        string memory;
        require(_input >= 0, "invalid uint8");
        require(_input <= 255, "invalid uint8");
        return "Input is UInt8";
    }
    // Defining function to use require statement
    function Odd(uint _input) public view returns(bool){
        require(_input % 2 != 0);
        return true;
    }
}

```

Deploy the contract requireStatement-requireF.sol.

```

// Solidity program to demonstrate require
// statement
pragma solidity ^0.4.24;

// Creating a contract
contract requirement {
    // Defining function to check input
    function eventExample(uint _input) public view returns(string memory){
        string memory;
        require(_input >= 0, "invalid uint8");
        require(_input <= 255, "invalid uint8");
        return "Input is UInt8";
    }
    // Defining function to use require statement
    function Odd(uint _input) public view returns(bool){
        require(_input % 2 != 0);
        return true;
    }
}

```

Perform the CheckInput and odd

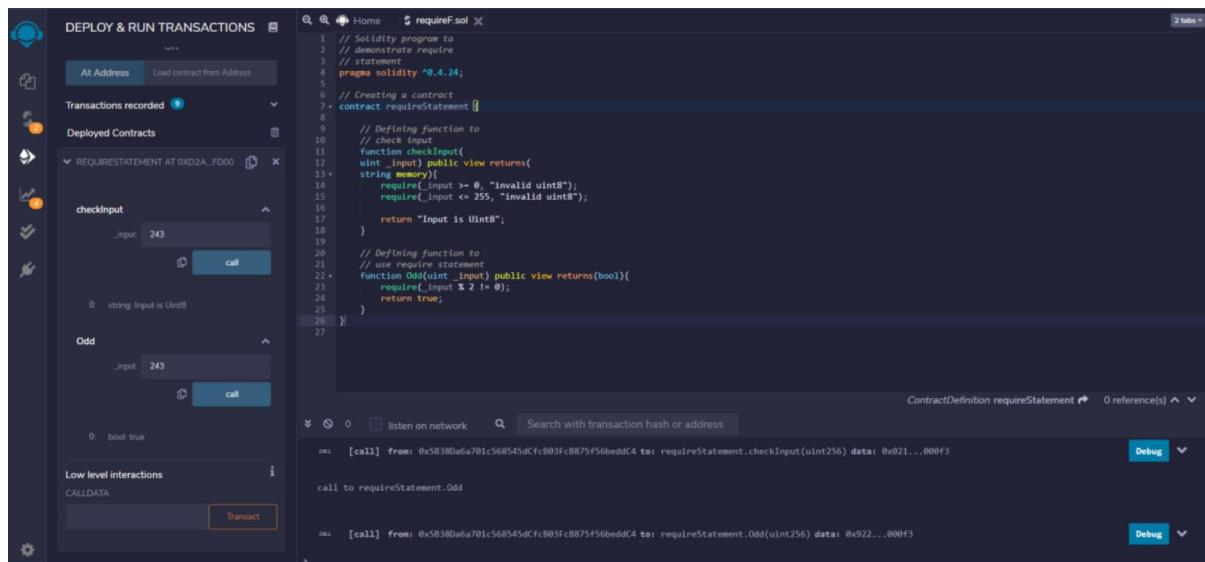
Code :

```
// Solidity program to
// demonstrate require
// statement
pragma solidity ^0.4.24;

// Creating a contract contract requireStatement {
// Defining function to
// check input
function checkInput(uint _input) public view returns( string memory){
    require(_input >= 0, "invalid uint8");
    require(_input <= 255, "invalid uint8"); return "Input is Uint8";
}

// Defining function to
// use require statement
function Odd(uint _input) public view returns(bool){
require(_input % 2 != 0);
return true;
}
```

OUTPUT:



Practical 5

Aim:

Install Hyperledger Fabric and Composer. Deploy and execute the application.

To install Hyperledger Fabric and Composer, there are some prerequisites that are required for blockchain application development.

You must have Ubuntu 18.04 LTS 64-bit as your Operating System.

Prerequisites are as follows:

- 1) cURL: latest version
- 2) Docker and Docker Compose: 17.06.2-ce or greater
- 3) Go: version 1.12x
- 4) Node.js: version 8 (8.94 or higher) or 10

Step 1: Log in as —root|| user.

In the terminal, you must have logged in as your username and hostname.

You need to log in as the —root|| user.

Command: `username@hostname:~$ sudo -i`

This command will ask you your username password to change user as root. Enter the password and you be logged is as —root@hostname||

Step 2: Update the package list

Command: `sudo apt-get update`

cURL

Step 3: Install cURL

Command: sudo apt install curl

Docker

Step 4: Install required packages for Docker

Command: sudo apt-get install apt-transport-https ca-certificates gnupg-agent software-properties-common

Step 5: Add Docker's official GPG key

Command: curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

Step 6: Setup the Docker stable repository

Command: sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable"

Step 7: Install the latest version of Docker Engine

Command: sudo apt-get install docker-ce docker-ce-cli containerd.io

Step 8: Add user to Docker Group

Command: sudo usermod -aG docker \$USER

Step 9: Verify the Docker Engine

Command: docker run hello-world

Docker Compose

Step 10: Download the latest version of Docker Compose

Command: sudo curl -L

"https://github.com/docker/compose/releases/download/1.25.5/docker-compose-\$(uname -s)-

\$(uname -m)" -o /usr/local/bin/docker-compose

Step 11: Apply Executable Permissions to the binary

Command: sudo chmod +x /usr/local/bin/docker-compose

Install Go

Step 12: Download the TAR file.

Command: curl -O https://storage.googleapis.com/golang/go1.12.9.linux-amd64.tar.gz

Step 13: Extract the File.

Command: tar -xvf go1.12.9.linux-amd64.tar.gz

Step 14: Update the environment variable Command: nano ~/.profile

Step 15: Add the below in the .profile file

```
export GOPATH=$HOME/go export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin Press
```

Ctrl+X, Enter Y and click Enter to come out of the file.

Step 16: Save the .profile file and load the updated environment variables.

Command: source ~/.profile

Install the Node.js and NPM

Step 17: Install Node.js Repository

Command: curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash –

Step 18: Install nodejs

Command: sudo apt-get install -y nodejs

Hyperledger Fabric

Install Hyperledger Fabric Samples, Binaries and Docker Images

Step 19: Before downloading the Fabric binaries, create a directory for e.g. hyperledger in your

home path.

Command: mkdir hyperledger cd hyperledger/

Step 20: Download Fabric v1.4.7

Command: curl -sSL http://bit.ly/2ysbOFE | bash -s -- 1.4.7 1.4.7 0.4.20

Step 21: Update the environment variable

Command: nano ~/.profile

Step 22: Add the fabric bin path to the PATH variable. Please make sure to use the path where you have downloaded fabric binaries.

```
export PATH=/home/$USER/hyperledger/fabric-samples/bin:$PATH
```

Press Ctrl+X, Enter Y and click Enter to come out of the file.

Step 23: Save the .profile file and load the updated environment variables.

Command: source ~/.profile

Testing the Hyperledger Fabric installation

Step 24: Change directory to the —first-network| directory

Command: cd /hyperledger/fabric-samples/first-network

Step 25: Use the script to bring up the network.

Command: ./byfn.sh up

```

root@ubuntu:~/hyperledger/fabric-samples/first-network/
containing directory /hyperledger/fabric-samples/first-network# ./byfn.sh up
starting for channel 'mychannel' with CLI timeout of '100' seconds and CLI delay of '3' seconds
Continue? [Y/n] Y
peer=peer0.org1.example.com
LOCAL_VERSION=1.4.7
DOCKER_IMAGE_VERSION=1.4.4
=====
===== WARNING =====
Some of the peer and docker images are
out of sync. This may cause problems.
=====
/rust/cryptogen
/rust/hyperledger/fabric-samples/first-network/../bin/cryptogen
=====
===== Generate certificates using cryptogen tool =====
cryptogen generate --config=./crypto-config.yaml
org1.example.com
+ ./seed
+ set +x
generate CCP files for Org1 and Org2
/rust/hyperledger/fabric-samples/first-network/../bin/configtxgen
=====
===== Generating Orderer Genesis Block =====
configtxgen -profile TwoOrgsOrdererGenesis -channelID byfn-sys-channel -role=orderer -file ./channel-artifacts/genesis.block
=====
===== Generating Peer genesis block =====
configtxgen -profile TwoOrgsChannel -channelID mychannel -role=peer -file ./channel-artifacts/channel.tx
=====
===== Generating channel configuration Transaction 'channel.tx' =====
configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/channel.tx -channelID mychannel
=====
===== Configuration file: ./channel-artifacts/channel.tx.yaml =====
root@ubuntu:~/hyperledger/fabric-samples/first-network#

```

```

6925ca32c85 hyperledger/fabric-tools:latest    "/bin/bash"      1 second ago   Up Less than a second
6f8e2145453f hyperledger/fabric-peer:latest     "peer node start" 3 seconds ago   Up Less than a second   0.0.0.0:318
69f520d39c hyperledger/fabric-peer:latest     "peer node start" 3 seconds ago   Up 2 seconds        0.0.0.0:398
b119403c55b hyperledger/fabric-peer:latest     "peer node start" 3 seconds ago   Up 2 seconds        0.0.0.0:378
1ab15fb1d98 hyperledger/fabric-orderer:latest   "orderer"        5 seconds ago   Up 1 second         0.0.0.0:70
1f95ab645c1 hello-world                         "/hello"          4 days ago    Exited (0) 4 days ago
9557343d747 hello-world                         "/hello"          4 days ago    Exited (0) 4 days ago
1961f699998 hello-world                         "/hello"          4 days ago    Exited (0) 4 days ago

```



Build your first network (BWN) end-to-end test.

```

channel name : mychannel
creating channel...
peer channel create -o orderer.example.com:7050 -c mychannel -f ./channel-artifacts/channel.tx --tls true --cafile /opt
on/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
+ res=0
+ set +x
2021-05-19 12:27:54.952 UTC [chaincodeCmd] InitChaincodeFunction -> INFO 001 Endorser and orderer connections initialized
2021-05-19 12:27:54.953 UTC [chaincodeCmd] resultBlock -> INFO 001 Received block: #0
=====
===== Channel 'mychannel' created =====
having all peers join the channel...
peer channel join -b mychannel.block
+ res=0
+ set +x
2021-05-19 12:27:58.337 UTC [chaincodeCmd] InitChaincodeFunction -> INFO 001 Endorser and orderer connections initialized
2021-05-19 12:27:58.338 UTC [chaincodeCmd] execQuery -> INFO 001 Successfully submitted proposal to join channel
=====
===== peer0.org1 joined channel 'mychannel' =====

```

```

Installing chaincode on peer1.org2...
+ peer chaincode install -n mycc -v 1.0 -l golang -p github.com/chaincode/chaincode_example02/go/
+ res=0
+ set +x
2021-05-19 12:29:29.818 UTC [chaincodeCmd] checkChaincodeInstall -> INFO 001 Using default escc
2021-05-19 12:29:29.818 UTC [chaincodeCmd] checkChaincodeInstall -> INFO 001 Using default vscc
2021-05-19 12:29:30.217 UTC [chaincodeCmd] install -> INFO 001 Installed remotely response:<status:200 payload:"OK">
=====
===== Chaincode is installed on peer1.org2 =====
Querying chaincode on peer1.org2...
===== Querying on peer1.org2 on channel 'mychannel'... =====
Attempting to Query peer1.org2 ... 3 secs
+ peer chaincode query -C mychannel -n mycc -c '{"Args":["query","a"]}'
+ res=0
+ set +x
2021-05-19 12:29:30.217 UTC [chaincodeCmd] query -> INFO 001 Query successful on peer1.org2 on channel 'mychannel'
=====
===== All GOOD, BYFN execution completed =====

```

Step 26: Verify the Docker containers

Command: docker ps -a

Step 27: Bring down the network.

Command: ./byfn.sh down

Hyperledger Composer

Step1: For new Droplets, always set the locale (choose en_US.UTF-8 if in doubt) and do apt update/upgrade.

Command: sudo dpkg-reconfigure locales; sudo apt-get update; sudo apt-get upgrade

Step 2: Add another user —playground®, using root will have all sorts of problems during installation, please make sure you use user —playground® to do it. Just follow the prompts to enter the info. You will need to give it a password which you will need later on.

Command: sudo adduser playground

Step 3: Set permissions for the new —playground® user account.

Command: sudo usermod -aG sudo playground

Step 4: Log in as —playground®, we will use this account to do the installation.

Command: su – playground

Step 5: Install the prerequisites by getting and running the script from github. It will ask for the password of —playground® account to proceed.

Command: curl -O <https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh>; chmod u+x prereqs-ubuntu.sh; ./prereqs-ubuntu.sh

Step 6: Log out of —playground® and log back in to proceed. Most of the time, you will ignore these kinds of steps on Linux, but this time, please do it, it won't work otherwise.

Command: exit; su – playground;

Step 7: Install components needed for running Hyperledger Composer.

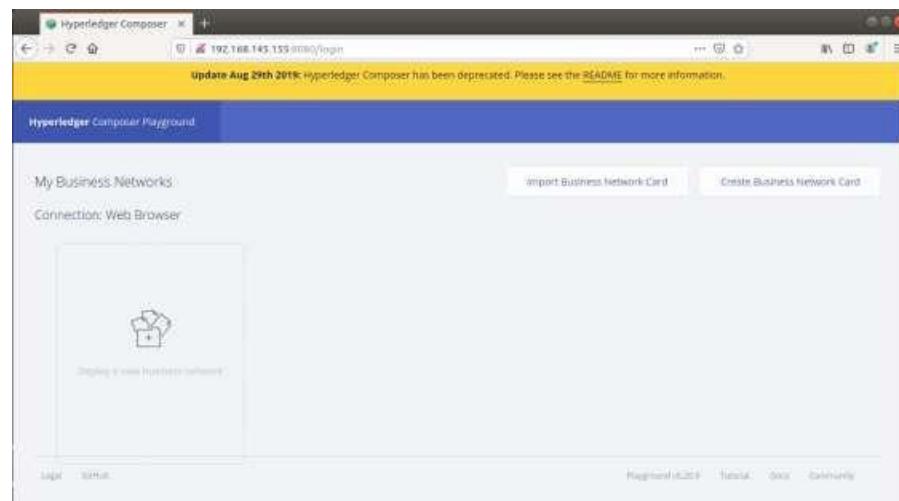
Command: npm install -g composer-cli composer-rest-server generator-hyperledgercomposer; yo composer-playground

Step 8: Start Composer, then you can access it via using your web-browser to <http://<Your IP Address>:8080>.

Command: composer-playground

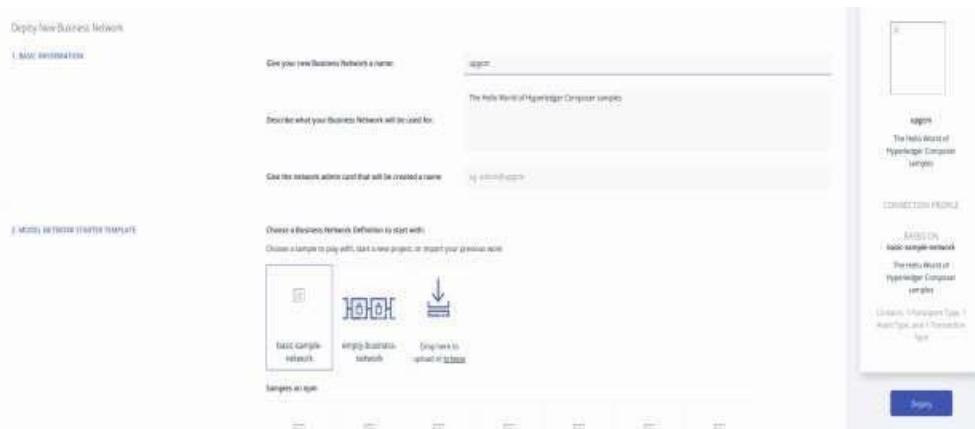
```
playground@ubuntu:~$ composer-playground
2021-05-19T12:50:32.126Z INFO    :LoadModule           :loadModule()
Loading composer-wallet-filesystem from /home/playground/.nvm/versions/node/v8.17.0/lib/node_modules/composer-playground/node_modules/composer-wallet-filesystem {}
2021-05-19T12:50:32.622Z INFO    :PlaygroundAPI       :createServer()
Playground API started on port 8080 {}$
```

This means the service has started and you can access the composer.



Webpages would like this.

Step 9: Click on Deploy a New Business Network

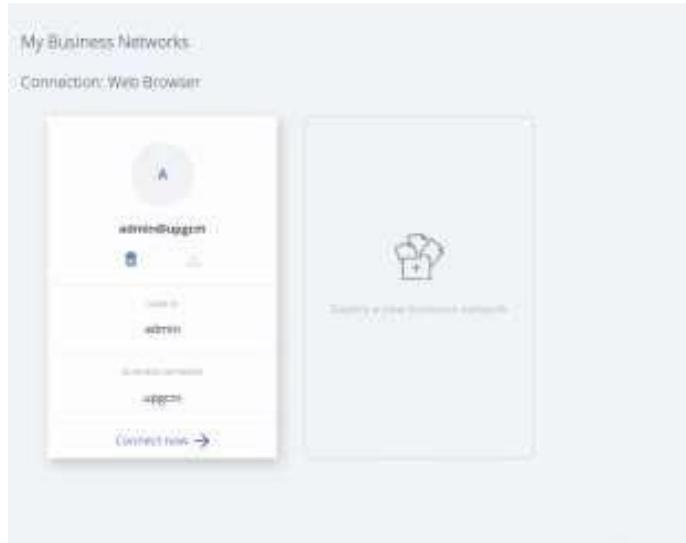


Give your new Business Network a name, choose basic-sample-network and click on

Deploy

Step 10: Connect the Network

Click on Connect Now



Step 11: Deploy the changes which are default made.

Click on Deploy Changes



Steps 12: Test the Application

Go to Test Tab

The screenshot shows the 'Participant registry for org.example.basic.SampleParticipant' page. On the left, there's a sidebar with 'Participants', 'SampleParticipant', 'Assets', 'SimpleAsset', and 'Transactions'. The main area has a table with columns 'ID' and 'Data'. A message at the bottom says 'This registry is empty! To create resources in this registry click create new at the top of this page.' There's also a 'Submit Transaction' button.

Step 13: Create New Participant and New Asset in SampleParticipant and SampleAsset respectively.

Two side-by-side 'Create New' dialog boxes. The left one is for 'Create New Participant' with JSON data:

```

1 {
2   "$class": "org.example.basic.SampleParticipant",
3   "participantId": "9000",
4   "firstName": "",
5   "lastName": ""
}

```

The right one is for 'Create New Asset' with JSON data:

```

1 {
2   "$class": "org.example.basic.SampleAsset",
3   "assetId": "4678",
4   "owner": "resource:org.example.basic.SampleParticipant#5099",
5   "value": ""
}

```

Both dialogs have 'Optional Properties' checkboxes, 'Generate Random Data' buttons, and 'Cancel'/'Create New' buttons.

You'll be able to see the entry in the respective section

Two screenshots of the blockchain interface showing the results of the creation. The top one is the 'Participant registry for org.example.basic.SampleParticipant' page, showing a single entry for ID 9000 with the JSON data from the previous step. The bottom one is the 'Asset registry for org.example.basic.SampleAsset' page, showing a single entry for ID 4678 with the JSON data from the previous step.

Note down the Asset ID. Here it is 4678

You can see the transaction of Adding Participant and Adding Asset in the All Transaction Section.

Date, Time	Entry Type	Participant	
2021-05-19, 06:24:08	AddAsset	admin (NetworkAdmin)	view record
2021-05-19, 06:23:00	AddParticipant	admin (NetworkAdmin)	view record

Step 14: Submit the transaction using the Asset ID.

Change the Asset ID in the 2nd line at last and submit.

Submit Transaction

Transaction Type: SampleTransaction

JSON Data Preview

```

1  {
2    "$class": "org.example.basic.SampleTransaction",
3    "asset": "resource:org.example.basic.SampleAsset@0",
4    "newValue": ""
5  }

```

Optional Properties:

just need quick test data? [Generate Random Data](#)

[Cancel](#) [Submit](#)

Transaction gets recorded in All Transaction Section

Date, Time	Entry Type	Participant	
2021-05-19, 06:32:41	SampleTransaction	admin (NetworkAdmin)	view record

You can view record and see that there is an event in it

Historian Record

Transaction Events (1)

org.example.basic.SampleEvent#dc12b2a8-6e67-4a49-b839-f7b7f76d5b1b#0

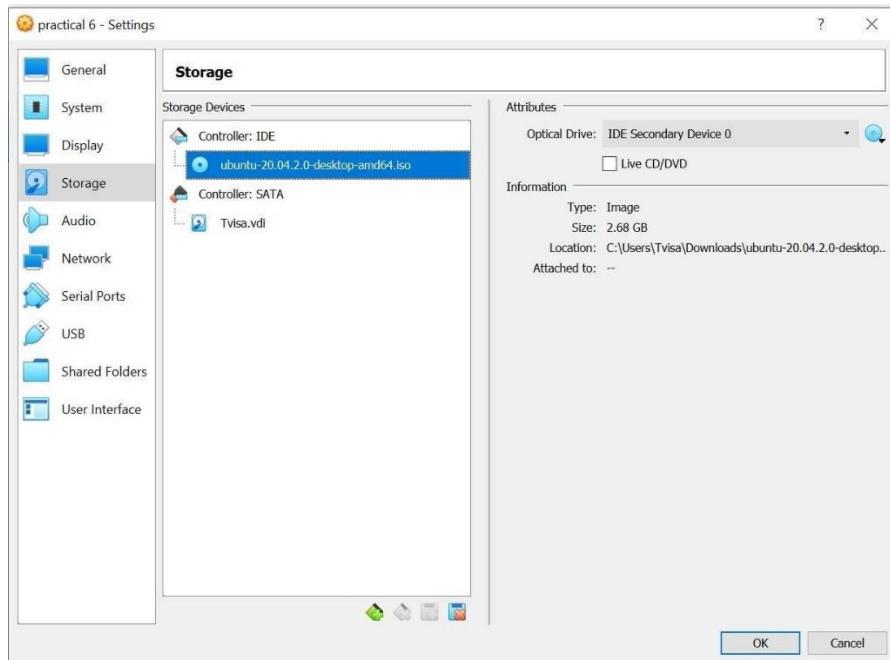
```
1  {
2    "$class": "org.example.basic.SampleEvent",
3    "asset": "resource:org.example.basic.SampleAsset#4678",
4    "oldValue": "",
5    "newValue": "",
6    "eventId": "dc12b2a8-6e67-4a49-b839-f7b7f76d5b1b#0",
7    "timestamp": "2021-05-19T13:32:41.517Z"
8 }
```

Practical 6

Aim: Write a Program to demonstrate mining of Ether.

Steps:

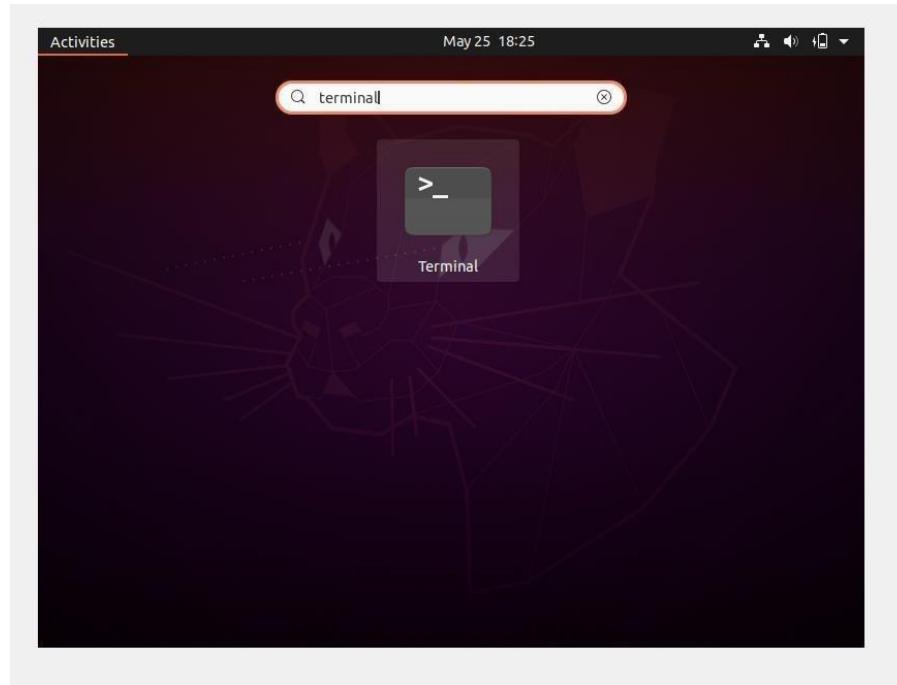
Step 1: Firstly we need to install Virtual Box. Once done with the installation process, create the Virtual Machine. After creating Virtual Machine, your machine will be created but some settings are required. Go to the Settings option as you can see below.



Step 2: Select the Virtual Machine, here we are going to select practical 6 machine and click on start.



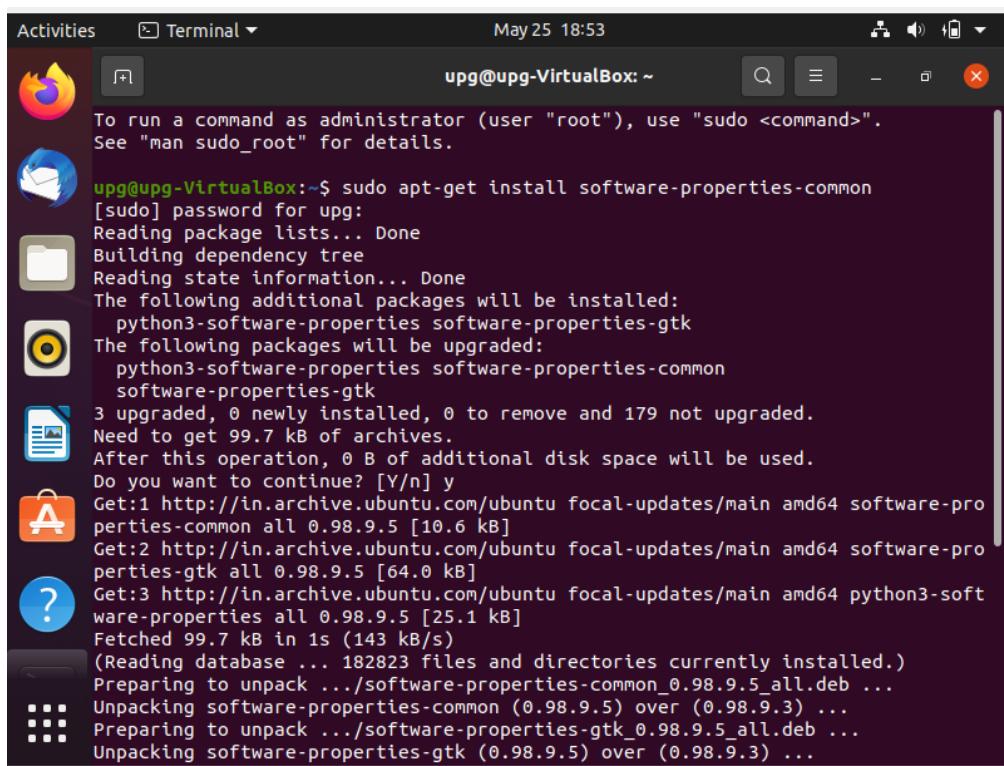
Step 3: Once the machine is on you need to go the search bar and type terminal



Step 4: After opening the terminal , you need to type the following command.

- sudo apt-get install software-properties-common

This software provides an abstraction of the used apt repositories. It allows you to easily manage your distribution and independent software vendor software sources. plus the DBUS backends to do the same via the Software and Updates GUI

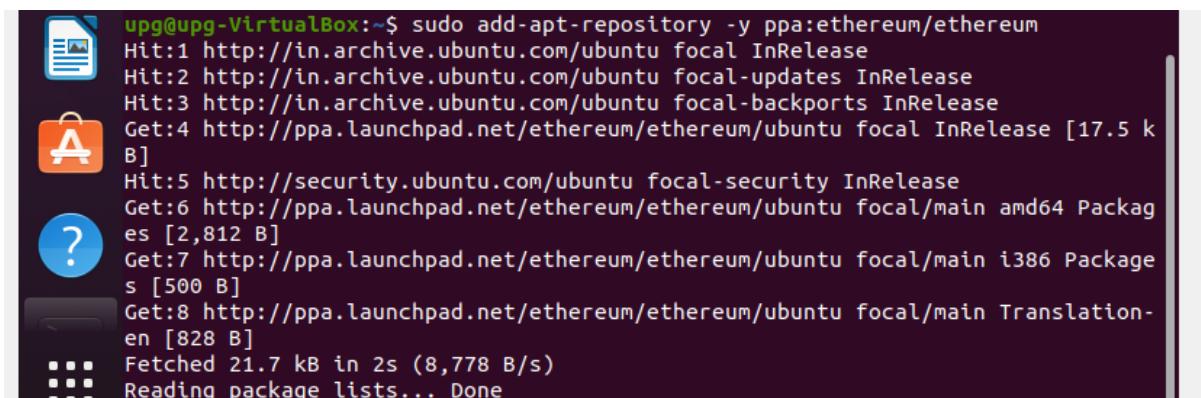


```
Activities Terminal May 25 18:53
upg@upg-VirtualBox:~$ sudo apt-get install software-properties-common
[sudo] password for upg:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties software-properties-gtk
The following packages will be upgraded:
  python3-software-properties software-properties-common
  software-properties-gtk
3 upgraded, 0 newly installed, 0 to remove and 179 not upgraded.
Need to get 99.7 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 software-properties-common all 0.98.9.5 [10.6 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 software-properties-gtk all 0.98.9.5 [64.0 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-software-properties all 0.98.9.5 [25.1 kB]
Fetched 99.7 kB in 1s (143 kB/s)
(Reading database ... 182823 files and directories currently installed.)
Preparing to unpack .../software-properties-common_0.98.9.5_all.deb ...
Unpacking software-properties-common (0.98.9.5) over (0.98.9.3) ...
Preparing to unpack .../software-properties-gtk_0.98.9.5_all.deb ...
Unpacking software-properties-gtk (0.98.9.5) over (0.98.9.3) ...
```

After the typing of the command, you need enter the password. Step 5: Type the second command as shown below:

- sudo add-apt-repository -y ppa:ethereum/Ethereum

add-apt-repository adds a PPA to your list of sources, (importing the GPG public key of the PPA automatically), so that Ubuntu knows to look for updates from that PPA as well as from the official Ubuntu sources. Usually this is used by developers to provide updates more quickly than in the official Ubuntu repositories.

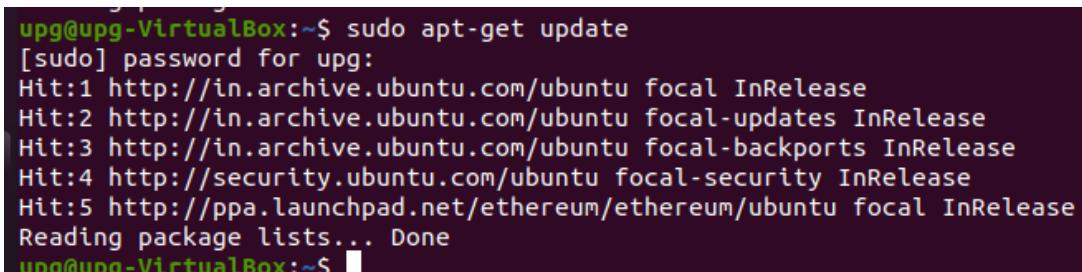


```
upg@upg-VirtualBox:~$ sudo add-apt-repository -y ppa:ethereum/ethereum
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://ppa.launchpad.net/ethereum/ethereum/ubuntu focal InRelease [17.5 kB]
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:6 http://ppa.launchpad.net/ethereum/ethereum/ubuntu focal/main amd64 Packages [2,812 B]
Get:7 http://ppa.launchpad.net/ethereum/ethereum/ubuntu focal/main i386 Packages [500 B]
Get:8 http://ppa.launchpad.net/ethereum/ethereum/ubuntu focal/main Translation-en [828 B]
Fetched 21.7 kB in 2s (8,778 B/s)
Reading package lists... Done
```

Step 6: After done with the above command its time to update.

- sudo apt-get update

Is used to download package information from all configured sources.



```
upg@upg-VirtualBox:~$ sudo apt-get update
[sudo] password for upg:
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 http://ppa.launchpad.net/ethereum/ethereum/ubuntu focal InRelease
Reading package lists... Done
upg@upg-VirtualBox:~$
```

Once again you need to type password as you can clearly see above.

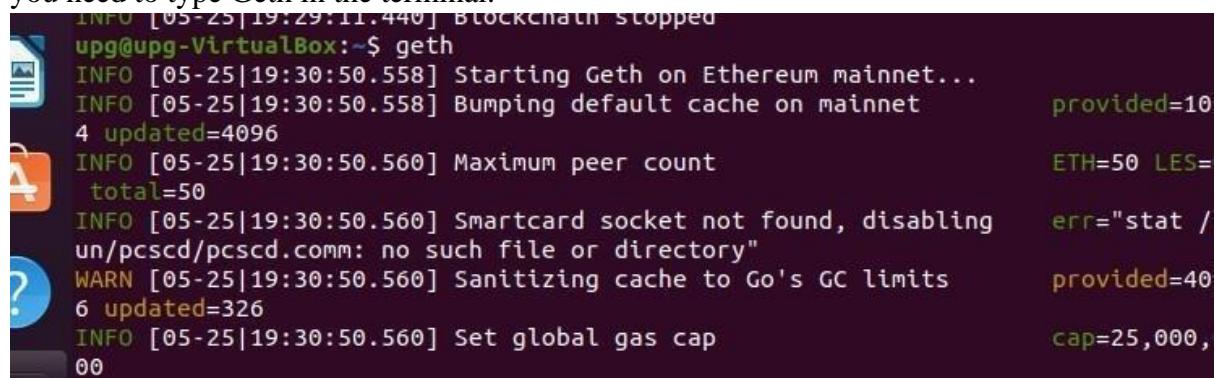
Step 7: Its time to install Ethereum.

- sudo apt-get install ethereum

This command will help us to download ethereum on our ubuntu,

```
upg@upg-VirtualBox:~$ sudo apt-get install ethereum
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  abi-gen bootnode clef evm geth puppeth rlpdump
The following NEW packages will be installed:
  abi-gen bootnode clef ethereum evm geth puppeth rlpdump
0 upgraded, 8 newly installed, 0 to remove and 179 not upgraded.
Need to get 31.6 MB of archives.
After this operation, 112 MB of additional disk space will be used.
```

Step 8: On Completing the installation process of Ethereum you need to run Geth. So here you need to type Geth in the terminal.



```
INFO [05-25|19:29:11.440] Blockchain stopped
upg@upg-VirtualBox:~$ geth
INFO [05-25|19:30:50.558] Starting Geth on Ethereum mainnet...
INFO [05-25|19:30:50.558] Bumping default cache on mainnet      provided=10
4 updated=4096
INFO [05-25|19:30:50.560] Maximum peer count                  ETH=50 LES=
total=50
INFO [05-25|19:30:50.560] Smartcard socket not found, disabling   err="stat /
un/pcscd/pcscd.comm: no such file or directory"
WARN [05-25|19:30:50.560] Sanitizing cache to Go's GC limits      provided=40
6 updated=326
INFO [05-25|19:30:50.560] Set global gas cap                  cap=25,000,
00
```

This will go on forever so Press Control + C to get the synchronization stop and you will be dropped back at your same old command-line prompt.

Step 9: To restart Geth with the console, type the following command:

- geth

If you already have a mist running and synchronized , you can tell Geth to use Mist's Node to connect by starting Geth via the following command.This is possible by typing following command:

- geth attach

After typing this command you can call console and attach one after the other.This is possible when your geth is fully synchronized.As result we can use the Javascript console in Geth right away.

Step 10: So your Geth Client should be running with the concole enabled, giving you the command prompt.

We will be creating an account by using the Javascript API call , but you need to keep in mind that the geth synchronization needs to be completely synchronized. So in Console type:

- personal.newAccount("your_new_account_password_here") Keep in mind to replace the text between the quotes with the password. Step 11:

After creating an account you will be returned with the public key which will be displayed in green color.

By typing:

- personal.listAccounts

It is possible for you to see all your accounts in the console.

Step 12: You can do all the above things in other way that is by using flags. Here we can use Geth Command line to launch Geth with certain flags. In order to get started with this start Geth on testnet by typing following command:

- geth - -testnet

Process will be same but here it will be happening on testnet. Press Ctrl+C to stop.

Step 13: Geth will not begin mining automatically you will have to command it to start or stop mining. In order to begin with mining on the main network, open a new terminal and enter JavaScript console by typing the following command:

- geth console

Nodes Synchronization will begin , but quickly return a command prompt and you can start by entering commands as Geth works in the background.

Step 14: You need to tell your node the Ethereum address for receiving your mining payments. This can be done by typing the following command in the console:

mine.setTherbase(eth.accounts[your_address_here]) Remember to replace quotes with your address.

Step 15: Start the Mining Process using the command:

- miner.start()

Mining process will begin and if you find block your payment will be received at the addresses you set above.

You can stop this process by typing:

- miner.stop()

```

root@gayatri-Aspire-E5-573G:/home/gayatri/Downloads# ssh -i neo.pem root@18.217.7.80
The authenticity of host '18.217.7.80 (18.217.7.80)' can't be established.
ECDSA key fingerprint is SHA256:Ik0/hYJscMT3sJvDfhlfwQw0bH098gv5JwKDBbNUv6g.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '18.217.7.80' (ECDSA) to the list of known hosts.
Please login as the user "ubuntu" rather than the user "root".

Connection to 18.217.7.80 closed.
root@gayatri-Aspire-E5-573G:/home/gayatri/Downloads# ssh -i neo.pem root@18.217.7.80
Please login as the user "ubuntu" rather than the user "root".

^CConnection to 18.217.7.80 closed.
root@gayatri-Aspire-E5-573G:/home/gayatri/Downloads# ssh -i neo.pem ubuntu@18.217.7.80
=====
|   |
|_(_ / Deep Learning AMI (Ubuntu 16.04) Version 43.0
|__|_|__|
=====

Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-1128-aws x86_64)

Please use one of the following commands to start the required environment with the framework of your choice:
for AWS MX 1.7 (+Keras2) with Python3 (CUDA 10.1 and Intel MKL-DNN) _____ source activate mxnet_p36
for AWS MX 1.8 (+Keras2) with Python3 (CUDA + and Intel MKL-DNN) _____ source activate mxnet_latest_p37
for AWS MX(+Amazon Elastic Inference) with Python3 _____ source activate amazonei_mxnet_p36
for AWS MX(+AWS Neuron) with Python3 _____ source activate aws_neuron_mxnet_p36
for TensorFlow(+Keras2) with Python3 (CUDA + and Intel MKL-DNN) _____ source activate tensorflow_p37
for TensorFlow 2(+Keras2) with Python3 (CUDA 10.1 and Intel MKL-DNN) _____ source activate tensorflow2_p36
for TensorFlow 2.3 with Python3 (CUDA + and Intel MKL-DNN) _____ source activate tensorflow2_latest_p37
for TensorFlow(+Amazon Elastic Inference) with Python3 _____ source activate amazonei_tensorflow_p36
for TensorFlow 2(+Amazon Elastic Inference) with Python3 _____ source activate amazonei_tensorflow2_p36
for TensorFlow(+AWS Neuron) with Python3 _____ source activate aws_neuron_tensorflow_p36
for PyTorch 1.4 with Python3 (CUDA 10.1 and Intel MKL) _____ source activate pytorch_p36
for PyTorch 1.7.1 with Python3.7 (CUDA 11.1 and Intel MKL) _____ source activate pytorch_latest_p37
for PyTorch (+AWS Neuron) with Python3 _____ source activate aws_neuron_pytorch_p36
for PyTorch 1.3.1 with(+Amazon Elastic Inference) with Python3 _____ source activate amazonei_pytorch_p36
for PyTorch 1.5.1 with(+Amazon Elastic Inference) with Python3 _____ source activate amazonei_pytorch_latest_p36
for base Python3 (CUDA 10.0) _____ source activate python3

```

```
# sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/trusty main" > /etc/apt/sources.list.d/dotnetdev.list'
```

```

To automatically activate base conda environment upon login, run: 'conda config --set auto_activate_base true'
Official Conda User Guide: https://docs.conda.io/projects/conda/en/latest/user-guide/
AWS Deep Learning AMI Homepage: https://aws.amazon.com/machine-learning/amis/
Developer Guide and Release Notes: https://docs.aws.amazon.com/dlami/latest/devguide/what-is-dlami.html
Support: https://forums.aws.amazon.com/forum.jspa?forumID=263
For a fully managed experience, check out Amazon SageMaker at https://aws.amazon.com/sagemaker
When using INF1 type instances, please update regularly using the instructions at: https://github.com/aws/aws-neuron-sdk/tree/master/release-notes
=====
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

26 packages can be updated.
2 of these updates are security updates.
To see these additional updates run: apt list --upgradable

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ip-172-31-21-20:~$ sudo su
root@ip-172-31-21-20:/home/ubuntu# sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/ trusty main" > /etc/apt/sources.list.d/dotnetdev.list'^C
root@ip-172-31-21-20:/home/ubuntu# sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/ trusty main" > /etc/apt/sources.list.d/dotnetdev.list'

```

```
# sudo apt-key adv --keyserver apt-mo.trafficmanager.net --recv-keys 417A0893
```

```
# sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/xenial main" > /etc/apt/sources.list.d/dotnetdev.list'
```

```
# sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 417A0893
```

```
etc/apt/sources.list.d/dotnetdev.list'
root@ip-172-31-21-20:/home/ubuntu# sudo apt-key adv --keyserver apt-mo.trafficmanager.net --recv-keys 417A0893
Executing: /tmp/tmp.uInPBbnwOM/gpg.1.sh --keyserver
apt-mo.trafficmanager.net
--recv-keys
417A0893
gpg: requesting key 417A0893 from hkp server apt-mo.trafficmanager.net
gpg: keyserver timed out
gpg: keyserver receive failed: keyserver error
root@ip-172-31-21-20:/home/ubuntu# sudo apt-key adv --keyserver apt-mo.trafficmanager.net --recv-keys 417A0893
Executing: /tmp/tmp.ZDq5adMs/gpg.1.sh --keyserver
apt-mo.trafficmanager.net
--recv-keys
417A0893
gpg: requesting key 417A0893 from hkp server apt-mo.trafficmanager.net
gpg: keyserver timed out
gpg: keyserver receive failed: keyserver error
root@ip-172-31-21-20:/home/ubuntu# sudo apt-key adv --keyserver apt-mo.trafficmanager.net --recv-keys 417A0893
Executing: /tmp/tmp.G6yk4BNNWG/gpg.1.sh --keyserver
apt-mo.trafficmanager.net --recv-keys
417A0893
gpg: can't open '417A0893'
root@ip-172-31-21-20:/home/ubuntu# sudo apt-key adv --keyserver apt-mo.trafficmanager.net --recv-keys 417A0893
Executing: /tmp/tmp.oyngJZuPSU/gpg.1.sh --keyserver
apt-mo.trafficmanager.net
--recv-keys
417A0893
gpg: requesting key 417A0893 from hkp server apt-mo.trafficmanager.net
gpg: keyserver timed out
gpg: keyserver receive failed: keyserver error
root@ip-172-31-21-20:/home/ubuntu# sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/ xenial main" > /etc/apt/sources.list.d/dotnetdev.list'
root@ip-172-31-21-20:/home/ubuntu# sudo apt-key adv --keyserver keyserver.ubuntu.com:80 --recv-keys 417A0893
Executing: /tmp/tmp.sOpjsqYhC0/gpg.1.sh --keyserver
hkp://keyserver.ubuntu.com:80
--recv-keys
417A0893
gpg: requesting key 417A0893 from hkp server keyserver.ubuntu.com
```

sudo apt-get update

```
# sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/trusty main" > /etc/apt/sources.list.d/dotnetdev.list'
```

sudo apt-get update

```
gpg: requesting key 417A0893 from hkp server keyserver.ubuntu.com
gpg: key 417A0893: public key "Totally Legit Signing Key <mallory@example.org>" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
root@ip-172-31-21-20:/home/ubuntu# sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease
Hit:6 https://apt.repos.neuron.amazonaws.com xenial InRelease
Hit:7 https://nvidia.github.io/libnvidia-container/stable/ubuntu16.04/amd64 InRelease
Hit:8 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu16.04/amd64 InRelease
Hit:9 https://nvidia.github.io/nvidia-docker/ubuntu16.04/amd64 InRelease
Get:10 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease [2,836 B]
Ign:10 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease
Get:11 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial/main amd64 Packages [11.4 kB]
Fetched 339 kB in 0s (399 kB/s)
Reading package lists... Done
W: GPG error: https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease: The following signatures couldn't be verified because t
he public key is not available: NO_PUBKEY B02C46DF417A0893
W: The repository 'https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease' is not signed.
N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-21-20:/home/ubuntu# sudo sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/dotnet-release/ trusty main" > /etc/apt/sources.list.d/dotnetdev.list'
root@ip-172-31-21-20:/home/ubuntu# sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease
Hit:6 https://nvidia.github.io/libnvidia-container/stable/ubuntu16.04/amd64 InRelease
Hit:7 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu16.04/amd64 InRelease
Hit:8 https://nvidia.github.io/nvidia-docker/ubuntu16.04/amd64 InRelease
Hit:9 https://apt.repos.neuron.amazonaws.com xenial InRelease
Get:10 https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease [2,836 B]
Ign:10 https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease
```

#_mkdir hwapp

cd hwapp

cd..

```
# sudo apt-get install dotnet-dev-1.0.4
```

```
# cd hwapp
```

```
# sudo apt-get update
```

```
Get:10 https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease [2,836 B]
Ign:10 https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease
Get:11 https://apt-mo.trafficmanager.net/repos/dotnet-release trusty/main amd64 Packages [9,370 B]
Fetched 337 kB in 0s (400 kB/s)
Reading package lists... Done
W: GPG error: https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease: The following signatures couldn't be verified because t
he public key is not available: NO_PUBKEY B02C46DF417A0893
W: The repository 'https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease' is not signed.
N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-21-20:/home/ubuntu# mkdir hwapp
root@ip-172-31-21-20:/home/ubuntu# cd hwapp/
root@ip-172-31-21-20:/home/ubuntu/hwapp# cd ..
root@ip-172-31-21-20:/home/ubuntu# sudo apt-get install dotnet-dev-1.0.4
Reading package lists... Done
Building dependency tree
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
  dotnet-dev-1.0.4 : Depends: dotnet-sharedframework-microsoft.netcore.app-1.1.2 but it is not going to be installed
                     Depends: dotnet-sharedframework-microsoft.netcore.app-1.0.5 but it is not going to be installed
E: Unable to correct problems, you have held broken packages.
root@ip-172-31-21-20:/home/ubuntu# cd hwapp/
root@ip-172-31-21-20:/home/ubuntu/hwapp# dotnet new xunit --framework netcoreapp1.1
dotnet: command not found
root@ip-172-31-21-20:/home/ubuntu/hwapp# sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease
Hit:6 https://apt.repos.neuron.amazonaws.com xenial InRelease
Hit:7 https://nvidia.github.io/libnvidia-container/stable/ubuntu16_04/amd64 InRelease
```

```
# sudo aptget update_allow-unauthenticated
```

```
# cd /etc/apt/sources.list
```

```
# cd /etc/apt/sources.list.d/
```

```
# ls
```

```
Hit:7 https://nvidia.github.io/libnvidia-container/stable/ubuntu16_04/amd64 InRelease
Hit:8 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu16_04/amd64 InRelease
Hit:9 https://nvidia.github.io/nvidia-docker/ubuntu16_04/amd64 InRelease
Get:10 https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease [2,836 B]
Ign:10 https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease
Fetched 328 kB in 0s (421 kB/s)
Reading package lists... Done
W: GPG error: https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease: The following signatures couldn't be verified because t
he public key is not available: NO_PUBKEY B02C46DF417A0893
W: The repository 'https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease' is not signed.
N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-21-20:/home/ubuntu/hwapp# sudo apt-get update --allow-unauthenticated
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease
Hit:6 https://nvidia.github.io/libnvidia-container/stable/ubuntu16_04/amd64 InRelease
Hit:7 https://apt.repos.neuron.amazonaws.com xenial InRelease
Hit:8 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu16_04/amd64 InRelease
Hit:9 https://nvidia.github.io/nvidia-docker/ubuntu16_04/amd64 InRelease
Get:10 https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease [2,836 B]
Ign:10 https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease
Fetched 328 kB in 0s (416 kB/s)
Reading package lists... Done
W: GPG error: https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease: The following signatures couldn't be verified because t
he public key is not available: NO_PUBKEY B02C46DF417A0893
W: The repository 'https://apt-mo.trafficmanager.net/repos/dotnet-release trusty InRelease' is not signed.
N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-21-20:/home/ubuntu/hwapp# cd /etc/apt/sources.list.
sources.list.d/ sources.list.save
root@ip-172-31-21-20:/home/ubuntu/hwapp# cd /etc/apt/sources.list.
sources.list.d/ sources.list.save
root@ip-172-31-21-20:/home/ubuntu/hwapp# cd /etc/apt/sources.list.d/
root@ip-172-31-21-20:/etc/apt/sources.list.d# ls
```

```
# rm dotnetdev.list

# sudo apt-get update

# cd ../../

# cd /home/ubuntu/hwapp/

# sudo apt-get install dotnet-dev-1.0.4

# dotnet new xunit --framework netcoreapp1.1
```

```
root@ip-1/2-31-21-20:/home/ubuntu/hwapp# cd /etc/apt/sources.list.d/
sources.list.d/ sources.list.save
root@ip-172-31-21-20:/home/ubuntu/hwapp# cd /etc/apt/sources.list.d/
sources.list.d/ sources.list.save
root@ip-172-31-21-20:/home/ubuntu/hwapp# cd /etc/apt/sources.list.d/
root@ip-172-31-21-20:/etc/apt/sources.list.d ls
dotnetdev.list      openjdk-r-ubuntu-ppa-xenial.list
neuron.list        openjdk-r-ubuntu-ppa-xenial.list.save
neuron.list.save    ubuntu-toolchain-r-ubuntu-test-xenial.list
nvidia-docker.list  ubuntu-toolchain-r-ubuntu-test-xenial.list.save
nvidia-docker.list.save
root@ip-172-31-21-20:/etc/apt/sources.list.d# rm dotnetdev.list
root@ip-172-31-21-20:/etc/apt/sources.list.d# sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease
Hit:6 https://nvidia.github.io/libnvidia-container/stable/ubuntu16.04/amd64 InRelease
Hit:7 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu16.04/amd64 InRelease
Hit:8 https://apt.repos.neuron.amazonaws.com xenial InRelease
Hit:9 https://nvidia.github.io/nvidia-docker/ubuntu16.04/amd64 InRelease
Fetched 325 kB in 0s (472 kB/s)
Reading package lists... Done
root@ip-172-31-21-20:/etc/apt/sources.list.d# cd ../../..
root@ip-172-31-21-20:# cd /home/ubuntu/hwapp/
root@ip-172-31-21-20:/home/ubuntu/hwapp# sudo apt-get install dotnet-dev-1.0.4
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package dotnet-dev-1.0.4
E: Couldn't find any package by glob 'dotnet-dev-1.0.4'
E: Couldn't find any package by regex 'dotnet-dev-1.0.4'
root@ip-172-31-21-20:/home/ubuntu/hwapp# ^C
root@ip-172-31-21-20:/home/ubuntu/hwapp# dotnet new xunit --framework netcoreapp1.1

Welcome to .NET Core!
-----
```

```
# ls
```

```
# dotnet restore hwapp.csproj
```

```
Welcome to .NET Core!
-----
Learn more about .NET Core @ https://aka.ms/dotnet-docs. Use dotnet --help to see available commands or go to https://aka.ms/dotnet-cli-docs.
Telemetry
-----
The .NET Core tools collect usage data in order to improve your experience. The data is anonymous and does not include command-line arguments.
The data is collected by Microsoft and shared with the community.
You can opt out of telemetry by setting a DOTNET_CLI_TELEMETRY_OPTOUT environment variable to 1 using your favorite shell.
You can read more about .NET Core tools telemetry @ https://aka.ms/dotnet-cl-telemetry.
Configuring...
-----
A command is running to initially populate your local package cache, to improve restore speed and enable offline access. This command will take up to a minute to complete and will only happen once.
Decompressing 100% 5648 ms
Expanding 100% 32844 ms
Getting ready...
Content generation time: 70.9277 ms
The template "xUnit Test Project" created successfully.
root@ip-172-31-21-20:/home/ubuntu/hwapp# ls
hwapp.csproj  UnitTest1.cs
root@ip-172-31-21-20:/home/ubuntu/hwapp# dotnet restore hwapp.csproj
Restoring packages for /home/ubuntu/hwapp/hwapp.csproj...
Installing System.Xml.XmlSerializer 4.0.11.
Installing System.Private.DataContractSerialization 4.1.1.
Installing System.Runtime.Serialization.Json 4.0.2.
Installing System.Diagnostics.TraceSource 4.0.0.
Installing System.Diagnostics.TextWriterTraceListener 4.0.0.
Installing System.ComponentModel.EventBasedAsync 4.0.11.
Installing System.Xml.XPath.XmlDocument 4.0.1.
Installing xunit.abstractions 2.0.1.
Installing Microsoft.TestPlatform.ObjectModel 15.0.0.
Installing xunit.extensibility.execution 2.2.0.
Installing xunit.extensibility.core 2.2.0.
Installing Microsoft.TestPlatform.TestHost 15.0.0.
Installing Microsoft.TestPlatform.ObjectModel 11.0.0.
Installing Microsoft.Extensions.DependencyInjection 1.1.0.
```

```
# ls
```

```

Installing Microsoft.TestPlatform.ObjectModel 11.0.0.
Installing Microsoft.Extensions.DependencyModel 1.1.0.
Installing Microsoft.DotNet.PlatformAbstractions 1.1.0.
Installing xunit.core 2.2.0.
Installing xunit.assert 2.2.0.
Installing Microsoft.NET.Test.Sdk 15.0.0.
Installing xunit.runner.visualstudio 2.2.0.
Installing xunit 2.2.0.
Generating MSBuild file /home/ubuntu/hwapp/obj/hwapp.csproj.nuget.g.props.
Generating MSBuild file /home/ubuntu/hwapp/obj/hwapp.csproj.nuget.g.targets.
Writing lock file to disk. Path: /home/ubuntu/hwapp/obj/project.assets.json
Restore completed in 2.38 sec for /home/ubuntu/hwapp/hwapp.csproj.

NuGet Config files used:
  /root/.nuget/NuGet.Config

Feeds used:
  https://api.nuget.org/v3/index.json

Installed:
  20 package(s) to /home/ubuntu/hwapp/hwapp.csproj
root@ip-172-31-21-20:/home/ubuntu/hwapp# ls
hwapp.csproj  obj  UnitTest1.cs
root@ip-172-31-21-20:/home/ubuntu/hwapp# dotnet run
root@ip-172-31-21-20:/home/ubuntu/hwapp# cd ..
root@ip-172-31-21-20:/home/ubuntu# sudo apt-get install libleveldb-dev sqlite3 libssqlite3-dev libunwind8-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libleveldb-dev is already the newest version (1.18-5).
The following packages were automatically installed and are no longer required:
  cpp-7 liblai0 librados2 librbd1
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  libssqlite3-dev libunwind-dev libunwind8-dev sqlite3
0 upgraded, 4 newly installed, 0 to remove and 27 not upgraded.

```

git clone <https://github.com/neo-project/neo.cli>

git branch -a

```

libssqlite3-dev libunwind-dev libunwind8-dev sqlite3
0 upgraded, 4 newly installed, 0 to remove and 27 not upgraded.
Need to get 1,396 kB of archives.
After this operation, 6,358 kB of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libssqlite3-dev amd64 3.11.0-1ubuntu1.5 [509 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/main amd64 libunwind-dev amd64 1.1-4.1 [366 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/main amd64 libunwind8-dev amd64 1.1-4.1 [2,614 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 sqlite3 amd64 3.11.0-1ubuntu1.5 [518 kB]
Fetched 1,396 kB in 0s (6,806 kB/s)
Selecting previously unselected package libssqlite3-dev:amd64.
(Reading database ... 132298 files and directories currently installed.)
Preparing to unpack .../libssqlite3-dev_3.11.0-1ubuntu1.5_amd64.deb ...
Unpacking libssqlite3-dev:amd64 (3.11.0-1ubuntu1.5) ...
Selecting previously unselected package libunwind-dev.
Preparing to unpack .../libunwind-dev_1.1-4.1_amd64.deb ...
Unpacking libunwind-dev (1.1-4.1) ...
Selecting previously unselected package libunwind8-dev.
Preparing to unpack .../libunwind8-dev_1.1-4.1_amd64.deb ...
Unpacking libunwind8-dev (1.1-4.1) ...
Selecting previously unselected package sqlite3.
Preparing to unpack .../sqlite3_3.11.0-1ubuntu1.5_amd64.deb ...
Unpacking sqlite3 (3.11.0-1ubuntu1.5) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up libssqlite3-dev:amd64 (3.11.0-1ubuntu1.5) ...
Setting up libunwind-dev (1.1-4.1) ...
Setting up libunwind8-dev (1.1-4.1) ...
Setting up sqlite3 (3.11.0-1ubuntu1.5) ...
root@ip-172-31-21-20:/home/ubuntu# git clone https://github.com/neo-project/neo-cli
Cloning into 'neo-cli'...
remote: Enumerating objects: 2139, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 2139 (delta 23), reused 4 (delta 2), pack-reused 2094
Receiving objects: 100% (2139/2139), 696.08 KiB | 0 bytes/s, done.
Resolving deltas: 100% (1437/1437), done.
Checking connectivity... done.
root@ip-172-31-21-20:/home/ubuntu# git branch -a

```

#mkdir blockchain

cd blockchain/

git clone <https://github.com/neo-project/neo-cli>

#ls

cd neo-cli/

```
# git status
```

```
# git branch -a
```

```
# git checkout v3.0
```

```
root@ip-172-31-21-20:/home/ubuntu# git branch -a
fatal: Not a git repository (or any of the parent directories): .git
root@ip-172-31-21-20:/home/ubuntu# mkdir blockchain
root@ip-172-31-21-20:/home/ubuntu# cd blockchain/
root@ip-172-31-21-20:/home/ubuntu/blockchain# git clone https://github.com/neo-project/neo-cli
Cloning into 'neo-cli'...
remote: Enumerating objects: 2139, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 2139 (delta 23), reused 3 (delta 2), pack-reused 2094
Receiving objects: 100% (2139/2139), 695.93 KiB | 0 bytes/s, done.
Resolving deltas: 100% (1436/1436), done.
Checking connectivity... done.
root@ip-172-31-21-20:/home/ubuntu/blockchain# ls
neo-cli
root@ip-172-31-21-20:/home/ubuntu/blockchain# git branch -a
fatal: Not a git repository (or any of the parent directories): .git
root@ip-172-31-21-20:/home/ubuntu/blockchain# git branch
fatal: Not a git repository (or any of the parent directories): .git
root@ip-172-31-21-20:/home/ubuntu/blockchain# git checkout v3.0
fatal: Not a git repository (or any of the parent directories): .git
root@ip-172-31-21-20:/home/ubuntu/blockchain# ls
neo-cli
root@ip-172-31-21-20:/home/ubuntu/blockchain# cd neo-cli/
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/attach-asset
  remotes/origin/master
  remotes/origin/master-2.x
  remotes/origin/reorder-transfer-args
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# git checkout v3.0
error: pathspec 'v3.0' did not match any file(s) known to git.
```

```
# wget https://packages.microsoft.com/config/ubuntu/16.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
```

```
# sudo dpkg -I packages-microsoft-prod.deb
```

```
# dotnet restore
```

```
#cd neo-cli/
```

```
remotes/origin/reorder-transfer-args
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# git checkout v3.0
error: pathspec 'v3.0' did not match any file(s) known to git.
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/attach-asset
  remotes/origin/master
  remotes/origin/master-2.x
  remotes/origin/reorder-transfer-args
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# ls
CHANGELOG_2.x.md Dockerfile LICENSE neo-cli Neo.ConsoleService neo-gui neo-node.sln NuGet.Config README.md tests
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# git checkout head
error: pathspec 'head' did not match any file(s) known to git.
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# cd ..
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# wget https://packages.microsoft.com/config/ubuntu/16.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
2021-05-10 16:34:07 - https://packages.microsoft.com/config/ubuntu/16.04/packages-microsoft-prod.deb
Resolving packages.microsoft.com (packages.microsoft.com)... 13.90.21.104
Connecting to packages.microsoft.com (packages.microsoft.com)|13.90.21.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2438 (2.4K) [application/octet-stream]
Saving to: 'packages-microsoft-prod.deb'

packages-microsoft-prod.deb      100%[=====] 2.38K --.-KB/s   in 0s
2021-05-18 16:34:08 (349 MB/s) - 'packages-microsoft-prod.deb' saved [2438/2438]

root@ip-172-31-21-20:/home/ubuntu/blockchain# sudo dpkg -i packages-microsoft-prod.deb
Selecting previously unselected package packages-microsoft-prod.
(Reading database ... 132369 files and directories currently installed.)
Preparing to unpack packages-microsoft-prod.deb ...
Unpacking packages-microsoft-prod (1.0-2) ...
Setting up packages-microsoft-prod (1.0-2) ...
root@ip-172-31-21-20:/home/ubuntu/blockchain# dotnet restore
MSB4103 : error MSB4103: Specify a project or solution file. The current working directory does not contain a project or solution file.
root@ip-172-31-21-20:/home/ubuntu/blockchain# cd neo-cli/
```

```
#dotnet restore
```

```
root@ip-172-31-21-20:/home/ubuntu/blockchain# cd neo-cli/
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# dotnet restore
Restoring packages for /home/ubuntu/blockchain/neo-cli/Neo.ConsoleService/Neo.ConsoleService.csproj...
  Installing System.Security.Principal.Windows 5.0.0.
  Installing System.Diagnostics.EventLog 5.0.0.
  Generating MSBuild file /home/ubuntu/blockchain/neo-cli/Neo.ConsoleService/obj/Neo.ConsoleService.csproj.nuget.g.props.
  Generating MSBuild file /home/ubuntu/blockchain/neo-cli/Neo.ConsoleService/obj/Neo.ConsoleService.csproj.nuget.g.targets.
  Writing lock file to disk. Path: /home/ubuntu/blockchain/neo-cli/Neo.ConsoleService/obj/project.assets.json
Restore completed in 3.04 sec for /home/ubuntu/blockchain/neo-cli/Neo.ConsoleService/Neo.ConsoleService.csproj.
Restoring packages for /home/ubuntu/blockchain/neo-cli/neo-cli/neo-cli.csproj...
  Installing System.Numerics.Vectors 4.5.0.
  Installing System.Buffers 4.5.1.
  Installing System.Runtime.CompilerServices.Unsafe 4.5.3.
  Installing System.Memory 4.5.4.
  Installing BouncyCastle.Nettle 1.8.8.
  Installing K4os.Compression.LZ4 1.2.6.
  Installing Neo.VM 3.0.0-rc2.
  Installing Microsoft.EntityFrameworkCore.Sqlite 5.0.5.
  Installing Akka 1.4.19.
  Installing Neo 3.0.0-rc2.
/usr/share/dotnet/sdk/1.0.4/NuGet.targets(97,5): error : Package Akka 1.4.19 is not compatible with net50 (.NETFramework,Version=v5.0). Package
e Akka 1.4.19 supports: netstandard2.0 (.NETStandard,Version=v2.0) [/home/ubuntu/blockchain/neo-cli/neo-node.sln]
/usr/share/dotnet/sdk/1.0.4/NuGet.targets(97,5): error : Package Microsoft.EntityFrameworkCore.Sqlite 5.0.5 is not compatible with net50 (.NET
Framework,Version=v5.0). Package Microsoft.EntityFrameworkCore.Sqlite 5.0.5 supports: netstandard2.1 (.NETStandard,Version=v2.1) [/home/ubuntu
/blockchain/neo-cli/neo-node.sln]
/usr/share/dotnet/sdk/1.0.4/NuGet.targets(97,5): error : One or more packages are incompatible with .NETFramework,Version=v5.0. [/home/ubuntu/
blockchain/neo-cli/neo-node.sln]
  Generating MSBuild file /home/ubuntu/blockchain/neo-cli/obj/neo-cli.csproj.nuget.g.props.
  Generating MSBuild file /home/ubuntu/blockchain/neo-cli/obj/neo-cli.csproj.nuget.g.targets.
  Writing lock file to disk. Path: /home/ubuntu/blockchain/neo-cli/neo-cli/obj/project.assets.json
Restore failed in 878.55 ms for /home/ubuntu/blockchain/neo-cli/neo-cli/neo-cli.csproj.
Restoring packages for /home/ubuntu/blockchain/neo-cli/neo-gui/neo-gui.csproj...
/usr/share/dotnet/sdk/1.0.4/NuGet.targets(97,5): error : Project Neo.CLI is not compatible with net50-windows (.NETFramework,Version=v5.0,Profile=windows
). Project Neo.CLI supports: net50 (.NETFramework,Version=v5.0) [/home/ubuntu/blockchain/neo-cli/neo-node.sln]
/usr/share/dotnet/sdk/1.0.4/NuGet.targets(97,5): error : One or more projects are incompatible with .NETFramework,Version=v5.0,Profile=windows
. [/home/ubuntu/blockchain/neo-cli/neo-node.sln]
```

```
(file=windows). Project Neo.CLI supports: net50 (.NETFramework,Version=v5.0) [/home/ubuntu/blockchain/neo-cli/neo-node.sln]
/usr/share/dotnet/sdk/1.0.4/NuGet.targets(97,5): error : One or more projects are incompatible with .NETFramework,Version=v5.0,Profile=windows
. [/home/ubuntu/blockchain/neo-cli/neo-node.sln]
  Generating MSBuild file /home/ubuntu/blockchain/neo-cli/neo-gui/obj/neo-gui.csproj.nuget.g.props.
  Generating MSBuild file /home/ubuntu/blockchain/neo-cli/neo-gui/obj/neo-gui.csproj.nuget.g.targets.
  Writing lock file to disk. Path: /home/ubuntu/blockchain/neo-cli/neo-gui/obj/project.assets.json
Restore failed in 1.63 ms for /home/ubuntu/blockchain/neo-cli/neo-gui/neo-gui.csproj.
Restoring packages for /home/ubuntu/blockchain/neo-cli/tests/Neo.ConsoleService.Tests/Neo.ConsoleService.Tests.csproj...
  Installing Microsoft.CodeCoverage 16.9.1.
  Installing Microsoft.NET.Test.Sdk 16.9.1.
  Installing MSTest.TestAdapter 2.2.3.
  Installing MSTest.TestFramework 2.2.3.
  Generating MSBuild file /home/ubuntu/blockchain/neo-cli/tests/Neo.ConsoleService.Tests/obj/Neo.ConsoleService.Tests.csproj.nuget.g.props.
  Generating MSBuild file /home/ubuntu/blockchain/neo-cli/tests/Neo.ConsoleService.Tests/obj/Neo.ConsoleService.Tests.csproj.nuget.g.targets.
  Writing lock file to disk. Path: /home/ubuntu/blockchain/neo-cli/tests/Neo.ConsoleService.Tests/obj/project.assets.json
Restore completed in 799.97 ms for /home/ubuntu/blockchain/neo-cli/tests/Neo.ConsoleService.Tests/Neo.ConsoleService.Tests.csproj.

Errors in /home/ubuntu/blockchain/neo-cli/neo-cli/neo-cli.csproj
  Package Akka 1.4.19 is not compatible with net50 (.NETFramework,Version=v5.0). Package Akka 1.4.19 supports: netstandard2.0 (.NETStandard
d,Version=v2.0)
  Package Microsoft.EntityFrameworkCore.Sqlite 5.0.5 is not compatible with net50 (.NETFramework,Version=v5.0). Package Microsoft.EntityFrameworkCore.Sqlite 5.0.5 supports: netstandard2.1 (.NETStandard,Version=v2.1)
  One or more packages are incompatible with .NETFramework,Version=v5.0.

Errors in /home/ubuntu/blockchain/neo-gui/neo-gui/neo-gui.csproj
  Project Neo.CLI is not compatible with net50-windows (.NETFramework,Version=v5.0,Profile=windows). Project Neo.CLI supports: net50 (.NET
Framework,Version=v5.0)
  One or more projects are incompatible with .NETFramework,Version=v5.0,Profile=windows.

NuGet Config files used:
  /home/ubuntu/blockchain/neo-cli/NuGet.Config
  /root/.nuget/NuGet.Config

Feeds used:
  https://www.myget.org/F/neo/api/v3/index.json
  https://api.nuget.org/v3/index.json
```

```
Installed:
  3 package(s) to /home/ubuntu/blockchain/neo-cli/Neo.ConsoleService/Neo.ConsoleService.csproj
  13 package(s) to /home/ubuntu/blockchain/neo-cli/neo-cli/neo-cli.csproj
  7 package(s) to /home/ubuntu/blockchain/neo-cli/tests/Neo.ConsoleService.Tests/Neo.ConsoleService.Tests.csproj
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# dotnet publish -c Release
Microsoft (R) Build Engine version 15.1.1012.6693
Copyright (C) Microsoft Corporation. All rights reserved.

/usr/share/dotnet/sdk/1.0.4/Microsoft.Common.CurrentVersion.targets(1111,5): error MSB3644: The reference assemblies for framework ".NETFramework,Version=v5.0" were not found. To resolve this, install the SDK or Targeting Pack for this framework version or retarget your application to a version of the framework for which you have the SDK or Targeting Pack installed. Note that assemblies will be resolved from the Global Assembly Cache (GAC) and will be used in place of reference assemblies. Therefore your assembly may not be correctly targeted for the framework you intend. [/home/ubuntu/blockchain/neo-cli/neo-cli.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: The "GenerateRuntimeConfigurationFiles" task failed unexpectedly. [/home/ubuntu/blockchain/neo-cli/neo-cli/neo-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: System.IO.DirectoryNotFoundException
  Could not find a part of the path '/home/ubuntu/blockchain/neo-cli/neo-gui/neo-gui/runtimeconfig.json'. [/home/ubuntu/blo
ckchain/neo-gui/neo-gui/neo-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at Interop.ThrowExceptionForIoEr
rorInfo(errorInfo, String path, Boolean isDirectory, Func<Boolean> errorRewriter) [/home/ubuntu/blockchain/neo-gui/neo-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at Interop.CheckIoSafeHandle(I
Handle handle, String path, Boolean isDirectory, Func<Boolean> errorRewriter) [/home/ubuntu/blockchain/neo-gui/neo-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at Microsoft.Win32.SafeHandles.Sa
fFileHandle.Open(String path, FileMode flags, Int32 mode) [/home/ubuntu/blockchain/neo-gui/neo-gui/neo-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at System.IO.UnixFileStream..ctor
  String path, FileMode mode, FileAccess access, FileShare share, Int32 bufferSize, FileOptions options, FileStream parent) [/home/ubuntu/block
chain/neo-gui/neo-gui/neo-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at System.IO.UnixFileSystem.Open(
  String fullPath, FileMode mode, FileAccess access, FileShare share, Int32 bufferSize, FileOptions options, FileStream parent) [/home/ubuntu/bl
ockchain/neo-gui/neo-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at System.IO.FileStream.Init(Stri
ng path, FileMode mode, FileAccess access, FileShare share, Int32 bufferSize, FileOptions options) [/home/ubuntu/blockchain/neo-gui/neo-gui/neo
-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at Microsoft.NET.Build.Tasks.Gene
rateRuntimeConfigurationFiles.WriteAllTextToFile(String fileName, Object value) [/home/ubuntu/blockchain/neo-gui/neo-gui/neo-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at Microsoft.NET.Build.Tasks.Gene
rateRuntimeConfigurationFiles.WriteRuntimeConfig(ProjectContext projectContext) [/home/ubuntu/blockchain/neo-gui/neo-gui/neo-gui.csproj]
```

```
# cd ..
```

```
# cd neo-cli/
```

```
#ls
```

```
# cd neo-cli/
```

```
#ls
```

```
#locate neo-cli.dll
```

```
# cd ../..
```

```
# sudo apt-get update; \
```

```
/usr/share/dotnet/sdk/1.0.4/Sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at Microsoft.NET.Build.Tasks.GenerateRuntimeConfigurationFiles.ExecuteCore() [/home/ubuntu/blockchain/neo-cli/neo-cli/neo-cli.csproj]
/usr/share/dotnet/sdk/1.0.4/Sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at Microsoft.NET.Build.Tasks.TaskBase.Execute() [/home/ubuntu/blockchain/neo-cli/neo-cli/neo-cli.csproj]
/usr/share/dotnet/sdk/1.0.4/Sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at Microsoft.Build.BackEnd.TaskExecutionHost.Microsoft.Build.BackEnd.ITaskExecutionHost.Execute() [/home/ubuntu/blockchain/neo-cli/neo-cli/neo-cli.csproj]
/usr/share/dotnet/sdk/1.0.4/Sdks/Microsoft.NET.Sdk/build/Microsoft.NET.Sdk.targets(129,5): error MSB4018: at Microsoft.Build.BackEnd.TaskBuilder.<ExecuteInstantiatedTask>d__25.MoveNext() [/home/ubuntu/blockchain/neo-cli/neo-cli/neo-cli.csproj]
/usr/share/dotnet/sdk/1.0.4/Sdks/Microsoft.NET.Sdk/build/Microsoft.NET.TargetFrameworkInference.targets(84,5): error : Cannot infer TargetFrameworkIdentifier and/or TargetFrameworkVersion from TargetFramework='net5.0-windows'. They must be specified explicitly. [/home/ubuntu/blockchain/neo-cli/neo-gui/neo-gui.csproj]
/usr/share/dotnet/sdk/1.0.4/Microsoft.Common.CurrentVersion.targets(1111,5): error MSB3044: The reference assemblies for framework ".NETFramework,Version=v5.0" were not found. To resolve this, install the SDK or Targeting Pack for this framework version or retarget your application to a version of the framework for which you have the SDK or Targeting Pack installed. Note that assemblies will be resolved from the Global Assembly Cache (GAC) and will be used in place of reference assemblies. Therefore your assembly may not be correctly targeted for the framework you intend. [/home/ubuntu/blockchain/neo-cli/Neo.ConsoleService.Tests/Neo.ConsoleService.Tests.csproj]
/usr/share/dotnet/sdk/1.0.4/Microsoft.Common.CurrentVersion.targets(1111,5): error MSB3044: The reference assemblies for framework ".NETFramework,Version=v5.0" were not found. To resolve this, install the SDK or Targeting Pack for this framework version or retarget your application to a version of the framework for which you have the SDK or Targeting Pack installed. Note that assemblies will be resolved from the Global Assembly Cache (GAC) and will be used in place of reference assemblies. Therefore your assembly may not be correctly targeted for the framework you intend. [/home/ubuntu/neo-gui/neo-gui.csproj]
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# cd ..
root@ip-172-31-21-20:/home/ubuntu/blockchain# dotnet /home/ubuntu/neo-cli/neo-cli/bin/Release/netcoreapp2.0/neo-cli.dll .
No executable found matching command "dotnet-/home/ubuntu/neo-cli/neo-cli/bin/Release/netcoreapp2.0/neo-cli.dll"
root@ip-172-31-21-20:/home/ubuntu/blockchain# cd neo-cli/
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# ls
CHANGELOG_2.x.md Dockerfile LICENSE neo-cli Neo.ConsoleService neo-gui neo-node.sln NuGet.Config README.md tests
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# cd neo-cli/
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli/neo-cli# ls
CLI config.json config.mainnet.json config.testnet.json Extensions.cs neo-cli.csproj obj Program.cs Settings.cs
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli/neo-cli# locate neo-cli.dll
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli/neo-cli# cd ../..
root@ip-172-31-21-20:/home/ubuntu/blockchain# sudo apt-get update; \
> sudo apt-get install -y apt-transport-https && \
> sudo apt-get update; \
```

```
root@ip-172-31-21-20:/home/ubuntu/blockchain# sudo apt-get update; \
> sudo apt-get install -y apt-transport-https && \
> sudo apt-get update && \
> sudo apt-get install -y dotnet-sdk-5.0
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease
Hit:6 https://nvidia.github.io/libnvidia-container/stable/ubuntu16.04/amd64 InRelease
Hit:7 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu16.04/amd64 InRelease
Hit:8 https://nvidia.github.io/nvidia-docker/ubuntu16.04/amd64 InRelease
Hit:9 https://apt.repos.neuron.amazonaws.com xenial InRelease
Get:10 https://packages.microsoft.com/ubuntu/16.04/prod xenial InRelease [4,003 B]
Get:11 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease [2,836 B]
Ign:11 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease
Get:12 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 Packages [289 kB]
Fetched 541 kB in 1s (501 kB/s)
Reading package lists... Done
W: GPG error: https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY B02C46DF417A0893
W: The repository 'https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease' is not signed.
N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use.
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  cpp-7 liblaloi1 librados2 librbd1
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  apt-transport-https
1 upgraded, 0 newly installed, 0 to remove and 28 not upgraded.
Need to get 26.6 kB of archives.
After this operation, 1,024 B of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apt-transport-https amd64 1.2.35 [26.6 kB]
Fetched 26.6 kB in 0s (0 B/s)
(Reading database ... 132374 files and directories currently installed.)
```

```

Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Fetched 26.6 kB in 0s (0 B/s)
(Reading database ... 132374 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.2.35_amd64.deb ...
Unpacking apt-transport-https (1.2.35) over (1.2.32ubuntu0.2) ...
Setting up apt-transport-https (1.2.35) ...
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease
Hit:6 https://apt.repos.neuron.amazonaws.com xenial InRelease
Hit:7 https://nvidia.github.io/libnvidia-container/stable/ubuntu16.04/amd64 InRelease
Hit:8 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu16.04/amd64 InRelease
Hit:9 https://nvidia.github.io/nvidia-docker/ubuntu16.04/amd64 InRelease
Hit:10 https://packages.microsoft.com/ubuntu/16.04/prod xenial InRelease
Get:11 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease [2,836 B]
Ign:11 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease
Fetched 328 kB in 0s (370 kB/s)
Reading package lists... Done
W: GPG error: https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease: The following signatures couldn't be verified because t
he public key is not available: NO_PUBKEY B02C46DF417A0893
W: The repository 'https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease' is not signed.
N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use.
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  _cpp-7 liblaloi_libraries2 librbbdi
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  aspnetcore-runtime-5.0 aspnetcore-targeting-pack-5.0 dotnet-apphost-pack-5.0 dotnet-host dotnet-hostfxr-5.0 dotnet-runtime-5.0
  dotnet-runtime-deps-5.0 dotnet-targeting-pack-5.0 netstandard-targeting-pack-2.1
The following NEW packages will be installed:
  aspnetcore-runtime-5.0 aspnetcore-targeting-pack-5.0 dotnet-apphost-pack-5.0 dotnet-hostfxr-5.0 dotnet-runtime-5.0 dotnet-runtime-deps-5.0
  dotnet-sdk-5.0 dotnet-targeting-pack-5.0 netstandard-targeting-pack-2.1
The following packages will be upgraded:
  dotnet-host
1 upgraded, 9 newly installed, 0 to remove and 27 not upgraded.
Need to get 94.1 MB of archives.
After this operation, 390 MB of additional disk space will be used.
Get:1 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 dotnet-runtime-deps-5.0 amd64 5.0.6-1 [2,642 B]
Get:2 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 dotnet-host amd64 5.0.6-1 [52.5 kB]
Get:3 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 dotnet-hostfxr-5.0 amd64 5.0.6-1 [140 kB]
Get:4 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 dotnet-runtime-5.0 amd64 5.0.6-1 [22.1 kB]
Get:5 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 aspnetcore-runtime-5.0 amd64 5.0.6-1 [6,086 kB]
Get:6 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 dotnet-targeting-pack-5.0 amd64 5.0.0-1 [2,086 kB]
Get:7 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 aspnetcore-targeting-pack-5.0 amd64 5.0.0-1 [1,316 kB]
Get:8 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 dotnet-apphost-pack-5.0 amd64 5.0.6-1 [3,412 kB]
Get:9 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 netstandard-targeting-pack-2.1 amd64 2.1.0-1 [1,476 kB]
Get:10 https://packages.microsoft.com/ubuntu/16.04/prod xenial/main amd64 dotnet-sdk-5.0 amd64 5.0.203-1 [57.5 MB]
Fetched 94.1 MB in 5s (17.3 MB/s)
Selecting previously unselected package dotnet-runtime-deps-5.0.
(Reading database ... 132374 files and directories currently installed.)
Preparing to unpack .../dotnet-runtime-deps-5.0_5.0.6-1_amd64.deb ...
Unpacking dotnet-runtime-deps-5.0 (5.0.6-1) ...
Preparing to unpack .../dotnet-host_5.0.6-1_amd64.deb ...
Unpacking dotnet-host (5.0.6-1) over (2.0.7-1) ...
Selecting previously unselected package dotnet-hostfxr-5.0.
Preparing to unpack .../dotnet-hostfxr-5.0_5.0.6-1_amd64.deb ...
Unpacking dotnet-hostfxr-5.0 (5.0.6-1) ...
Selecting previously unselected package dotnet-runtime-5.0.
Preparing to unpack .../dotnet-runtime-5.0_5.0.6-1_amd64.deb ...
Unpacking dotnet-runtime-5.0 (5.0.6-1) ...
Selecting previously unselected package aspnetcore-runtime-5.0.
Preparing to unpack .../aspnetcore-runtime-5.0_5.0.6-1_amd64.deb ...
Unpacking aspnetcore-runtime-5.0 (5.0.6-1) ...
Selecting previously unselected package dotnet-targeting-pack-5.0.
Preparing to unpack .../dotnet-targeting-pack-5.0_5.0.0-1_amd64.deb ...
Unpacking dotnet-targeting-pack-5.0 (5.0.0-1) ...
Selecting previously unselected package aspnetcore-targeting-pack-5.0.
Preparing to unpack .../aspnetcore-targeting-pack-5.0_5.0.0-1_amd64.deb ...
Unpacking aspnetcore-targeting-pack-5.0 (5.0.0-1) ...

Preparing to unpack .../dotnet-targeting-pack-5.0_5.0.0-1_amd64.deb ...
Unpacking dotnet-targeting-pack-5.0 (5.0.0-1) ...
Selecting previously unselected package aspnetcore-targeting-pack-5.0.
Preparing to unpack .../aspnetcore-targeting-pack-5.0_5.0.0-1_amd64.deb ...
Unpacking aspnetcore-targeting-pack-5.0 (5.0.0-1) ...
Selecting previously unselected package dotnet-apphost-pack-5.0.
Preparing to unpack .../dotnet-apphost-pack-5.0_5.0.6-1_amd64.deb ...
Unpacking dotnet-apphost-pack-5.0 (5.0.6-1) ...
Selecting previously unselected package netstandard-targeting-pack-2.1.
Preparing to unpack .../netstandard-targeting-pack-2.1_2.1.0-1_amd64.deb ...
Unpacking netstandard-targeting-pack-2.1 (2.1.0-1) ...
Selecting previously unselected package dotnet-sdk-5.0.
Preparing to unpack .../dotnet-sdk-5.0_5.0.203-1_amd64.deb ...
Unpacking dotnet-sdk-5.0 (5.0.203-1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up dotnet-runtime-deps-5.0 (5.0.6-1) ...
Setting up dotnet-host (5.0.6-1) ...
Setting up dotnet-hostfxr-5.0 (5.0.6-1) ...
Setting up dotnet-runtime-5.0 (5.0.6-1) ...
Setting up aspnetcore-runtime-5.0 (5.0.6-1) ...
Setting up dotnet-targeting-pack-5.0 (5.0.0-1) ...
Setting up aspnetcore-targeting-pack-5.0 (5.0.0-1) ...
Setting up dotnet-apphost-pack-5.0 (5.0.6-1) ...
Setting up netstandard-targeting-pack-2.1 (2.1.0-1) ...
Setting up dotnet-sdk-5.0 (5.0.203-1) ...
This software may collect information about you and your use of the software, and send that to Microsoft.
Please visit http://aka.ms/dotnet-cl-i-eula for more information.
Welcome to .NET!
-----
Learn more about .NET: https://aka.ms/dotnet-docs
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cl-i-docs

Telemetry
-----
The .NET tools collect usage data in order to help us improve your experience. It is collected by Microsoft and shared with the community. You
can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.

Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cl-i-telemetry
```

```
Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry
Configuring ...
-----
A command is running to populate your local package cache to improve restore speed and enable offline access. This command takes up to one minute to complete and only runs once.
root@ip-172-31-21-20:/home/ubuntu/blockchain# sudo apt-get update; \
> sudo apt-get install -y apt-transport-https && \
> sudo apt-get update && \
> sudo apt-get install -y aspnetcore-runtime-5.0
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/openjdk-7/ubuntu xenial InRelease
Hit:6 https://nvidia.github.io/libnvidiactl-container/stable/ubuntu16.04/amd64 InRelease
Hit:7 https://apt.repos.neuron.amazonaws.com xenial InRelease
Hit:8 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu16.04/amd64 InRelease
Hit:9 https://nvidia.github.io/nvidia-docker/ubuntu16.04/amd64 InRelease
Get:10 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease [2,836 B]
Hit:11 https://packages.microsoft.com/ubuntu/16.04/prod xenial InRelease
Ign:10 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease
Fetched 328 kB in 0s (364 kB/s)
Reading package lists... Done
W: GPG error: https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY B02C46DF417A0893
W: The repository 'https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease' is not signed.
N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use.
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree
Reading state information... Done
apt-transport-https is already the newest version (1.2.35).
The following packages were automatically installed and are no longer required:
  _cpp-7 liblai0 liblai0s2 liblrb0d1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
```

```
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/openjdk-7/ubuntu xenial InRelease
Hit:6 https://nvidia.github.io/libnvidiactl-container/stable/ubuntu16.04/amd64 InRelease
Hit:7 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu16.04/amd64 InRelease
Hit:8 https://apt.repos.neuron.amazonaws.com xenial InRelease
Hit:9 https://nvidia.github.io/nvidia-docker/ubuntu16.04/amd64 InRelease
Get:10 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease [2,836 B]
Ign:11 https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease
Fetched 328 kB in 0s (372 kB/s)
Reading package lists... Done
W: GPG error: https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY B02C46DF417A0893
W: The repository 'https://apt-mo.trafficmanager.net/repos/dotnet-release xenial InRelease' is not signed.
N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use.
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree
Reading state information... Done
aspnetcore-runtime-5.0 is already the newest version (5.0.6-1).
aspnetcore-runtime-5.0 set to manually installed.
The following packages were automatically installed and are no longer required:
  _cpp-7 liblai0 liblai0s2 liblrb0d1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
root@ip-172-31-21-20:/home/ubuntu/blockchain# sudo apt-get install -y dotnet-runtime-5.0
Reading package lists... Done
Building dependency tree
Reading state information... Done
dotnet-runtime-5.0 is already the newest version (5.0.6-1).
dotnet-runtime-5.0 set to manually installed.
The following packages were automatically installed and are no longer required:
  _cpp-7 liblai0 liblai0s2 liblrb0d1
Use 'sudo apt autoremove' to remove them.
```

#dotnet restore

cd neo-cli/

#ls

cd ../../

```
The following packages were automatically installed and are no longer required:
  _cpp-7 liblai0 liblai0s2 liblrb0d1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
root@ip-172-31-21-20:/home/ubuntu/blockchain# dotnet restore

Welcome to .NET 5.0!
-----
SDK Version: 5.0.203

Telemetry
-----
The .NET tools collect usage data in order to help us improve your experience. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.

Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Installed an ASP.NET Core HTTPS development certificate.
To trust the certificate run 'dotnet dev-certs https --trust' (Windows and macOS only).
Learn about HTTPS: https://aka.ms/dotnet-https

Write your first app: https://aka.ms/dotnet-hello-world
Find out what's new: https://aka.ms/dotnet-whats-new
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli

MSBUILD : error MSB1003: Specify a project or solution file. The current working directory does not contain a project or solution file.
root@ip-172-31-21-20:/home/ubuntu/blockchain# cd neo-cl/
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cl# dotnet restore
[usr/share/dotnet/sdk/5.0.203/Sdks/Microsoft.NET.Sdk/targets/Microsoft.NET.Sdk.FrameworkReferenceResolution.targets(63,5): error NETSDK1100: W
e need at least one project to build Windows desktop applications. [/home/ubuntu/blockchain/neo-cl/neo-gui/neo-gui.csproj]
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cl# ls
CHANGELOG.md  Dockerfile LICENSE neo-cl Neo.ConsoleService neo-gui neo-node.sln NuGet.Config README.md tests
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cl# cd ../..
root@ip-172-31-21-20:/home/ubuntu#
```

```
# cd neo
#mkdir neo
# cd neo/
# sudo apt-get install libleveldb-dev sqlite libsqlite3-dev libunwind8-dev
# git clone -b master-2.x https://github.com/neo-project/neo-node.git # ls
# cd neo-node/
# ls
#cd neo-cli
# ls
```

```
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# ls
CHANGELOG_2.x.md CHANGELOG.md Dockerfile LICENSE neo-cli Neo.ConsoleService neo-gui neo-node.sln NuGet.Config README.md tests
root@ip-172-31-21-20:/home/ubuntu/blockchain/neo-cli# cd ../..
root@ip-172-31-21-20:/home/ubuntu# cd neo
bash: cd: neo: No such file or directory
root@ip-172-31-21-20:/home/ubuntu# mkdir neo
root@ip-172-31-21-20:/home/ubuntu# cd neo/
root@ip-172-31-21-20:/home/ubuntu/neo# sudo apt-get install libleveldb-dev sqlite3 libsqlite3-dev libunwind8-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libleveldb-dev is already the newest version (1.18-5).
libunwind8-dev is already the newest version (1.1-4.1).
libsqlite3-dev is already the newest version (3.11.0-1ubuntu1.5).
sqlite3 is already the newest version (3.11.0-1ubuntu1.5).
The following packages were automatically installed and are no longer required:
  cpp-7 libalio1 librados2 librbd1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
root@ip-172-31-21-20:/home/ubuntu/neo# $ git clone -b master-2.x https://github.com/neo-project/neo-node.git
$: command not found
root@ip-172-31-21-20:/home/ubuntu/neo# git clone -b master-2.x https://github.com/neo-project/neo-node.git
Cloning into 'neo-node'...
remote: Enumerating objects: 2139, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 2139 (delta 23), reused 3 (delta 2), pack-reused 2094
Receiving objects: 100% (2139/2139), 695.93 KiB | 0 bytes/s, done.
Resolving deltas: 100% (1436/1436), done.
Checking connectivity... done.
root@ip-172-31-21-20:/home/ubuntu/neo# ls
neo-node
root@ip-172-31-21-20:/home/ubuntu/neo# cd neo-node/
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node# ls
CHANGELOG.md cl LICENSE neo-cli neo-cli.sln NuGet.Config README.md tests
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node# cd neo-cli
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# ls
config.json config.testnet.json neo-cli.csproj Properties protocol.mainnet.json Services Shell
```

```
# locate cli.dll
# dotnet restore
# dotnet publish -c release -r linux-x64
#ls
#cd din/
# ls
# cd release/
```

#ls

```
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node# ls
CHANGELOG.md cl LICENSE neo-cli neo-cli.sln NuGet.Config README.md tests
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node# cd neo-cli
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# ls
config.json config.testnet.json Properties protocol.mainnet.json Services Shell
config.mainnet.json Helper.cs Program.cs protocol.json protocol.testnet.json Settings.cs
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# locate cli.dll
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# dotnet restore
Determining projects to restore...
Restored /home/ubuntu/neo/neo-node/neo-cli/neo-cli.csproj (in 33.17 sec).
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# dotnet publish -c release -r linux-x64
Microsoft (R) Build Engine Version 16.9.0+57a23d249 For .NET
Copyright (C) Microsoft Corporation. All rights reserved.

Determining projects to restore...
Restored /home/ubuntu/neo/neo-node/neo-cli/neo-cli.csproj (in 9.44 sec).
/home/ubuntu/neo/neo-node/neo-cli/Services/ConsoleServiceBase.cs(300,34): warning CS0168: The variable 'ex' is declared but never used [/home/ubuntu/neo/neo-node/neo-cli/neo-cli.csproj]
neo-cli -> /home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/neo-cli.dll
neo-cli -> /home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/publish/
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# ls
bin config.mainnet.json obj Properties protocol.mainnet.json Services Shell
config.json config.testnet.json neo-cli.csproj Program.cs protocol.json protocol.testnet.json Settings.cs
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# d otnet /home/ubuntu/neo-cli/bin/Release/
d: command not found
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# dotnet /home/ubuntu/neo-cli/neo-cli/bin/Release/netcoreapp2.0/neo-cli.dll .
could not execute because the specified command or file was not found.
Possible reasons for this include:
* You misspelled a built-in dotnet command.
* You intended to execute a .NET program, but dotnet-/home/ubuntu/neo-cli/bin/Release/netcoreapp2.0/neo-cli.dll does not exist.
* You intended to run a global tool, but a dotnet-prefixed executable with this name could not be found on the PATH.
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# cd bin/
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin# ls
release
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin# cd release/
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release# ls
netcoreapp2.1
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release# cd netcoreapp2.1/
```

cd netcoreapp2.1/

ls

cd linux-x64

ls

cd publish

ls

```
netcoreapp2.1
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release# cd netcoreapp2.1/
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1# ls
linux-x64
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1# cd linux-x64/
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64# ls
config.json libhostfxr.so neo-cli.deps.json neo-cli.runtimeconfig.json protocol.testnet.json
config.mainnet.json libhostpolicy.so neo-cli.dll protocol.json publish
config.testnet.json neo-cli neo-cli.runtimeconfig.dev.json protocol.mainnet.json
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64# cd publish/
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/publish# ls
Akka.dll System.IO.FileSystem.AccessControl.dll
config.json System.IO.FileSystem.dll
config.mainnet.json System.IO.FileSystem.DriveInfo.dll
config.testnet.json System.IO.FileSystem.Primitives.dll
createdump System.IO.FileSystem.Watcher.dll
libclrjit.so System.IO.IsolatedStorage.dll
libcoreclr.so System.IO.MemoryMappedFiles.dll
libcoreclrtraceptprovider.so System.IO.Pipelines.dll
libddbshim.so System.IO.Pipes.AccessControl.dll
libe_sqlite3.so System.IO.Pipes.dll
libhostfxr.so System.IO.UnmanagedMemoryStream.dll
libhostpolicy.so System.Linq.dll
libmscordaccore.so System.Linq.Expressions.dll
libmscordbi.so System.Linq.Parallel.dll
libsspplugin.so System.Linq.Queryable.dll
libsos.so System.Memory.dll
Microsoft.AspNetCore.Connections.Abstractions.dll System.Native.a
Microsoft.AspNetCore.Hosting.Abstractions.dll System.Native.so
Microsoft.AspNetCore.Hosting.dll System.Net.dll
Microsoft.AspNetCore.Hosting.Server.Abstractions.dll System.Net.Http.dll
Microsoft.AspNetCore.Http.Abstractions.dll System.Net.HttpListener.dll
Microsoft.AspNetCore.Http.dll System.Net.Http.Native.a
Microsoft.AspNetCore.Http.Extensions.dll System.Net.Http.Native.so
Microsoft.AspNetCore.Http.Features.dll System.Net.Mail.dll
Microsoft.AspNetCore.ResponseCompression.dll System.Net.NameResolution.dll
Microsoft.AspNetCore.Server.Kestrel.Core.dll System.Net.NetworkInformation.dll
Microsoft.AspNetCore.Server.Kestrel.dll System.Net.Ping.dll
```

Microsoft.AspNetCore.Hosting.Server.Abstractions.dll	System.Net.Http.dll
Microsoft.AspNetCore.Http.Abstractions.dll	System.Net.HttpListener.dll
Microsoft.AspNetCore.Http.dll	System.Net.Http.Native.a
Microsoft.AspNetCore.Http.Extensions.dll	System.Net.Http.Native.so
Microsoft.AspNetCore.Http.Features.dll	System.Net.Mail.dll
Microsoft.AspNetCore.ResponseCompression.dll	System.Net.NameResolution.dll
Microsoft.AspNetCore.Server.Kestrel.Core.dll	System.Net.NetworkInformation.dll
Microsoft.AspNetCore.Server.Kestrel.dll	System.Net.Ping.dll
Microsoft.AspNetCore.Server.Kestrel.Https.dll	System.Net.Primitives.dll
Microsoft.AspNetCore.Server.Kestrel.Transport.Abstractions.dll	System.Net.Requests.dll
Microsoft.AspNetCore.Server.Kestrel.Transport.Sockets.dll	System.Net.Security.dll
Microsoft.AspNetCore.WebSockets.dll	System.Net.Security.Native.a
Microsoft.AspNetCore.WebUtilities.dll	System.Net.Security.Native.so
Microsoft.CSharp.dll	System.Net.ServicePoint.dll
Microsoft.Data.SQLite.dll	System.Net.Sockets.dll
Microsoft.EntityFrameworkCore.Abstractions.dll	System.Net.WebClient.dll
Microsoft.EntityFrameworkCore.dll	System.Net.WebHeaderCollection.dll
Microsoft.EntityFrameworkCore.Relational.dll	System.Net.WebProxy.dll
Microsoft.EntityFrameworkCore.Sqlite.dll	System.Net.WebSockets.Client.dll
Microsoft.Extensions.Caching.Abstractions.dll	System.Net.WebSockets.dll
Microsoft.Extensions.Caching.Memory.dll	System.Net.WebSockets.WebSocketProtocol.dll
Microsoft.Extensions.Configuration.Abstractions.dll	System.Numerics.dll
Microsoft.Extensions.Configuration.Binder.dll	System.Numerics.Vectors.dll
Microsoft.Extensions.Configuration.dll	System.ObjectModel.dll
Microsoft.Extensions.Configuration.EnvironmentVariables.dll	System.Private.CoreLib.dll
Microsoft.Extensions.Configuration.FileExtensions.dll	System.PrivateDataContractSerialization.dll
Microsoft.Extensions.Configuration.Json.dll	System.Private.Uri.dll
Microsoft.Extensions.DependencyInjection.Abstractions.dll	System.Private.Xml.dll
Microsoft.Extensions.DependencyInjection.dll	System.Private.Xml.Linq.dll
Microsoft.Extensions.DependencyModel.dll	System.Reflection.DispatchProxy.dll
Microsoft.Extensions.FileProviders.Abstractions.dll	System.Reflection.dll
Microsoft.Extensions.FileProviders.Physical.dll	System.Reflection.Emit.dll
Microsoft.Extensions.FileSystemGlobbing.dll	System.Reflection.Emit.Lightweight.dll
Microsoft.Extensions.Hosting.Abstractions.dll	System.Reflection.Extensions.dll
Microsoft.Extensions.Logging.Abstractions.dll	System.Reflection.Metadata.dll
Microsoft.Extensions.Logging.dll	System.Reflection.Primitives.dll
Microsoft.Extensions.ObjectPool.dll	System.Reflection.TypeExtensions.dll
Microsoft.Extensions.Options.dll	
Microsoft.Extensions.Primitives.dll	
Microsoft.Extensions.Primitives.dll	System.Reflection.TypeExtensions.dll
Microsoft.Net.Http.Headers.dll	System.Resources.Reader.dll
Microsoft.VisualBasic.dll	System.Resources.ResourceManager.dll
Microsoft.Win32.Primitives.dll	System.Resources.Writer.dll
mscorlib.dll	System.Runtime.CompilerServices.Unsafe.dll
neo-cli	System.Runtime.CompilerServices.VisualC.dll
neo-cli.deps.json	System.Runtime.dll
neo-cli.dll	System.Runtime.Extensions.dll
neo-cli.runtimeconfig.json	System.Runtime.Handles.dll
Neo.dll	System.Runtime.InteropServices.dll
Neo.WM.dll	System.Runtime.InteropServices.RuntimeInformation.dll
netstandard.dll	System.Runtime.InteropServices.WindowsRuntime.dll
Newtonsoft.Json.dll	System.Runtime.Loader.dll
protocol.json	System.Runtime.Numerics.dll
protocol.mainnet.json	System.Runtime.Serialization.dll
protocol.testnet.json	System.Runtime.Serialization.Formatters.dll
Remotion.Linq.dll	System.Runtime.Serialization.Json.dll
sosdocsunix.txt	System.Runtime.Serialization.Primitives.dll
SOS.NETCore.dll	System.Runtime.Serialization.Xml.dll
SQLitePCLRaw.batteries_green.dll	System.Security.AccessControl.dll
SQLitePCLRaw.batteries_v2.dll	System.Security.Claims.dll
SQLitePCLRaw.core.dll	System.Security.Cryptography.Algorithms.dll
SQLitePCLRaw.provider.e_sqlite3.dll	System.Security.Cryptography.Cng.dll
System.AppContext.dll	System.Security.Cryptography.Csp.dll
System.Buffers.dll	System.Security.Cryptography.Encoding.dll
System.Collections.Concurrent.dll	System.Security.Cryptography.Native.OpenSsl.a
System.Collections.dll	System.Security.Cryptography.Native.OpenSsl.so
System.Collections.Immutable.dll	System.Security.Cryptography.OpenSsl.dll
System.Collections.NonGeneric.dll	System.Security.Cryptography.Primitives.dll
System.Collections.Specialized.dll	System.Security.Cryptography.X509Certificates.dll
System.ComponentModel.Annotations.dll	System.Security.dll
System.ComponentModel.DataAnnotations.dll	System.Security.Permissions.dll
System.ComponentModel.dll	System.Security.Principal.dll
System.ComponentModel.EventBasedAsync.dll	System.Security.Principal.Windows.dll
System.ComponentModel.Primitives.dll	System.Security.SecureString.dll
System.ComponentModel.TypeConverter.dll	System.ServiceModel.Web.dll
System.Configuration.dll	System.ServiceProcess.dll
System.Configuration.dll	System.ServiceProcess.ServiceController.dll
Microsoft.Extensions.Primitives.dll	System.Reflection.TypeExtensions.dll
Microsoft.Net.Http.Headers.dll	System.Resources.Reader.dll
Microsoft.VisualBasic.dll	System.Resources.ResourceManager.dll
Microsoft.Win32.Primitives.dll	System.Resources.Writer.dll
Microsoft.Win32.Registry.dll	System.Runtime.CompilerServices.Unsafe.dll
mscorlib.dll	System.Runtime.CompilerServices.VisualC.dll
neo-cli	System.Runtime.dll
neo-cli.deps.json	System.Runtime.Extensions.dll
neo-cli.dll	System.Runtime.Handles.dll
neo-cli.runtimeconfig.json	System.Runtime.InteropServices.dll
Neo.dll	System.Runtime.InteropServices.RuntimeInformation.dll
Neo.WM.dll	System.Runtime.InteropServices.WindowsRuntime.dll
netstandard.dll	System.Runtime.Loader.dll
Newtonsoft.Json.dll	System.Runtime.Numerics.dll
protocol.json	System.Runtime.Serialization.dll
protocol.mainnet.json	System.Runtime.Serialization.Formatters.dll
protocol.testnet.json	System.Runtime.Serialization.Json.dll
Remotion.Linq.dll	System.Runtime.Serialization.Primitives.dll
sosdocsunix.txt	System.Runtime.Serialization.Xml.dll
SOS.NETCore.dll	System.Security.AccessControl.dll
SQLitePCLRaw.batteries_green.dll	System.Security.Claims.dll
SQLitePCLRaw.batteries_v2.dll	System.Security.Cryptography.Algorithms.dll
SQLitePCLRaw.core.dll	System.Security.Cryptography.Cng.dll
SQLitePCLRaw.provider.e_sqlite3.dll	System.Security.Cryptography.Csp.dll
System.AppContext.dll	System.Security.Cryptography.Encoding.dll
System.Buffers.dll	System.Security.Cryptography.Native.OpenSsl.a
System.Collections.Concurrent.dll	System.Security.Cryptography.Native.OpenSsl.so
System.Collections.dll	System.Security.Cryptography.OpenSsl.dll
System.Collections.Immutable.dll	System.Security.Cryptography.Primitives.dll
System.Collections.NonGeneric.dll	System.Security.Cryptography.X509Certificates.dll
System.Collections.Specialized.dll	System.Security.dll
System.ComponentModel.Annotations.dll	System.Security.Permissions.dll
System.ComponentModel.DataAnnotations.dll	System.Security.Principal.dll
System.ComponentModel.dll	System.Security.Principal.Windows.dll
System.ComponentModel.EventBasedAsync.dll	System.Security.SecureString.dll
System.ComponentModel.Primitives.dll	System.ServiceModel.Web.dll
System.ComponentModel.TypeConverter.dll	System.ServiceProcess.dll
System.Configuration.dll	System.ServiceProcess.ServiceController.dll

```
# locate libleveldb.dll
# cd ../../..
# cd ../..
# ls
# dotnet /home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/p
# dotnet /home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/neo-cli.dll Neo >
show state
```

```
System.Drawing.dll
System.Drawing.Primitives.dll
System.Dynamic.Runtime.dll
System.Globalization.Calendars.dll
System.Globalization.dll
System.Globalization.Extensions.dll
System.Globalization.Native.so
System.Interactive.Async.dll
System.IO.Compression.Brotli.dll
System.IO.Compression.dll
System.IO.Compression.FileSystem.dll
System.IO.Compression.Native.a
System.IO.Compression.Native.so
System.IO.Compression.ZipFile.dll
System.IO.dll
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/publish# locate libleveldb.dll
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/publish# cd ../../..
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli/bin/release# cd ../..
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# ls
bin config.mainnet.json Helper.cs obj Properties protocol.mainnet.json Services Shell
config.json config.testnet.json neo-cli.csproj Program.cs protocol.json protocol.testnet.json Settings.cs
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# dotnet /home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/p
protocol.json protocol.mainnet.json protocol.testnet.json publish/
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# dotnet /home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/neo-cli.dll
NEO-CLI Version: 2.12.2.0

neo> show state

block: 0/3102/4000 state:-1 connected: 8 unconnected: 151
  ip: 5.135.140.23      port: 10333    listen: 10333    height: 5299496
  ip: 52.76.5.84        port: 10333    listen: 10333    height: 7371658
  ip: 47.104.177.205   port: 10333    listen: 10333    height: 7371658
  ip: 13.90.195.175    port: 10333    listen: 10333    height: 7371657
  ip: 50.18.122.176    port: 10333    listen: 10333    height: 7371658
  ip: 18.136.178.96    port: 10333    listen: 10333    height: 7371658
  ip: 49.12.67.179     port: 10333    listen: 10333    height: 7371658
  ip: 65.21.105.228   port: 10333    listen: 10333    height: 7371658
```

```
neo> show state

block: 0/3102/4000 state:-1 connected: 8 unconnected: 151
  ip: 5.135.140.23      port: 10333    listen: 10333    height: 5299496
  ip: 52.76.5.84        port: 10333    listen: 10333    height: 7371658
  ip: 47.104.177.205   port: 10333    listen: 10333    height: 7371658
  ip: 13.90.195.175    port: 10333    listen: 10333    height: 7371657
  ip: 50.18.122.176    port: 10333    listen: 10333    height: 7371658
  ip: 18.136.178.96    port: 10333    listen: 10333    height: 7371658
  ip: 49.12.67.179     port: 10333    listen: 10333    height: 7371658
  ip: 65.21.105.228   port: 10333    listen: 10333    height: 7371658
```

```
# dotnet /home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/neo-cli.dll
Neo > create wallet wallet.db3
Neo > show gas
Neo > export key
```

```
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# dotnet /home/ubuntu/neo/neo-node/neo-cli/bin/release/netcoreapp2.1/linux-x64/neo-clidll
NEO-CLI Version: 2.12.2.0

neo> create wallet wallet.db3
password: *****
password: *****
address: APd5Z0xdfToRA2kNx6MvhYsJBFboGDEaEf
pubkey: 023d29dce586ce8593d1c5f4d2b653d9b697320e57edd7109b64f33ab03e6ffada
neo> show gas
unavailable: 0
available: 0
neo> export key
password: *****
Kzn1dvhZeM4VFUQmxFa29EVMZvGnr6PkPJSEozHypwEjgWcAzCix
neo> ^C
root@ip-172-31-21-20:/home/ubuntu/neo/neo-node/neo-cli# Connection to 18.217.7.80 closed by remote host.
Connection to 18.217.7.80 closed.
root@gayatri-Aspire-E5-573G:/home/gayatri/Downloads#
root@gayatri-Aspire-E5-573G:/home/gayatri/Downloads#
```

The screenshot shows the 'Open Wallet' section of the NEO Tracker website. It includes fields for entering a private key or a keystore file, and buttons for creating a keystore or going directly to the wallet.

Open Wallet

How would you like to access your wallet? *

Paste or type private key.

Keystore File

Private Key

Encrypted Key

UNLOCK

CREATE KEYSTORE

GO TO WALLET

Donate: [AK0VSYGluzmnkdiBaejuWeZvdvVEv01X](https://neotacker.io/wallet)

NEO Tracker © 2017-2021

<https://neotacker.io/wallet>

Twitter

Facebook

Practical 7

Aim: Demonstrate the running of the blockchain node.

The main purpose behind having a Blockchain application is to ensure the integrity of data and provide credibility to the network. These properties are maintained by sharing the same ledger amongst different systems, distributed globally.

The concept of Blockchain states that every block that contains data is connected cryptographically to the next block. If you change or modify any block, all the subsequent blocks have to be changed as their value or their hash depends on the previous block.

However, if this cryptographic chain of blocks was present at a single location, any hacker could change the value of all the blocks. In order to sustain the integrity of data, the Blockchain is a distributed network that shares the same copy of ledger or data among multiple systems.

This is the reason why every Blockchain network requires globally distributed systems to be truly decentralized.

If the ledger of any one system is changed, the ledger on the other systems acts as proof of the integrity of data. Therefore, the availability of these systems or Blockchain nodes is what allows it to be distributed and trustworthy. Without nodes, Blockchain is just a database secured with cryptographic hashing!

Regardless of the Blockchain being public or [private blockchain](#), a globally distributed network is essential for the perseverance of the data, achieved by saving the transactional record on the Blockchain nodes. The immutable record on nodes is the reason behind the unquestionable auditability of data or [transaction records on Blockchain](#).

Moreover, having different Blockchain nodes across the global network allows the network to be resilient towards any centralized attacks or even from any natural calamities.

Even if a whole nation gets destroyed due to any reason, all it needs is one Blockchain node to provide the Blockchain ledger to the network.

Practical 8

Aim: Demonstrate the use of Bitcoin Core API.

We are going to perform the practical in Debian Linux Operating System.

Step 1: Download Bitcoin Core API from following link <https://bitcoin.org/en/download>.

Step 2: Click on the Linux (tgz) and go to the location where file is downloaded.

Step 3: Unzip the file using the following command:

```
$ tar xzf bitcoin-0.21.1-x86_64-linux-gnu.tar.gz
```

Step 4: Install Bitcoin Core API using following command:

```
$ sudo install -m 0755 -o root -g root -t /usr/local/bin bitcoin-0.21.1/bin/*
```

Step 5: Bitcoin Core API is installed, now start the application by running following command:

```
$ /usr/local/bin/bitcoin-qt
```

Step 6: Select option —Use the default data directory|| and Click OK.

Step 7: Create a new Wallet by clicking on the button —Create a new wallet.

Step 8: Enter Wallet Name and then Click on —Create|| Button.

Step 9: Click on the Window -> Console.

Step 10: Enter following commands to get the desired output. There are many commands given in following link: <https://developer.bitcoin.org/reference/rpc/index.html>. We will use some of them.

getbalance - Returns the total available balance.

getbalances - Returns an object with all balances in BTC.

getblock - If verbosity is 0, returns a string that is serialized, hex-encoded data for block _hash‘.

getblockcount - Returns the height of the most-work fully-validated chain.

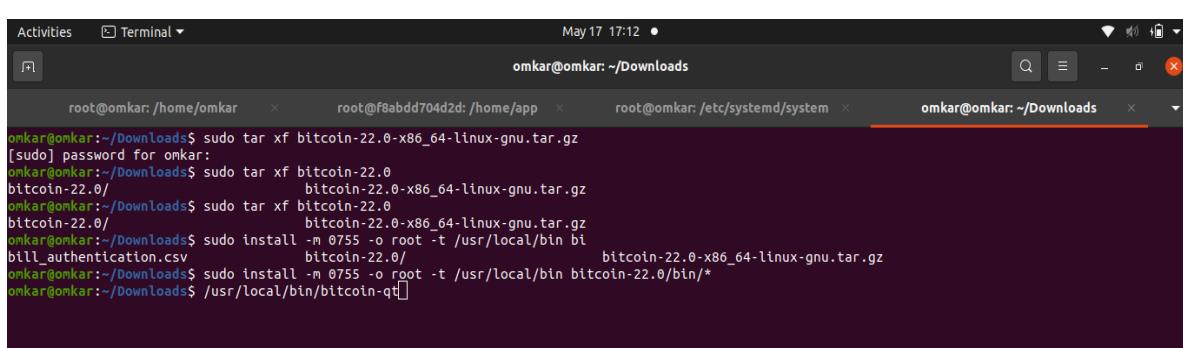
getmemoryinfo - Returns an object containing information about memory usage.

uptime - Returns the total uptime of the server (in seconds).

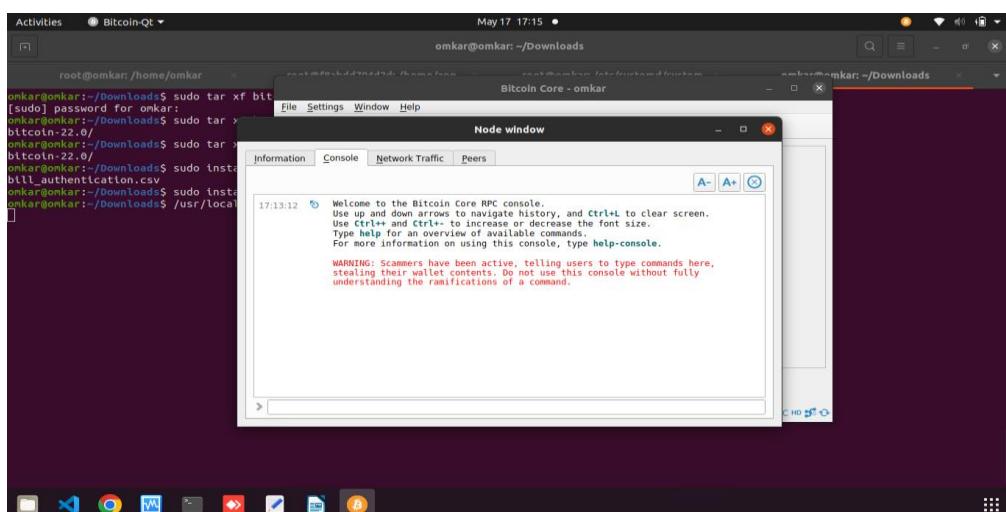
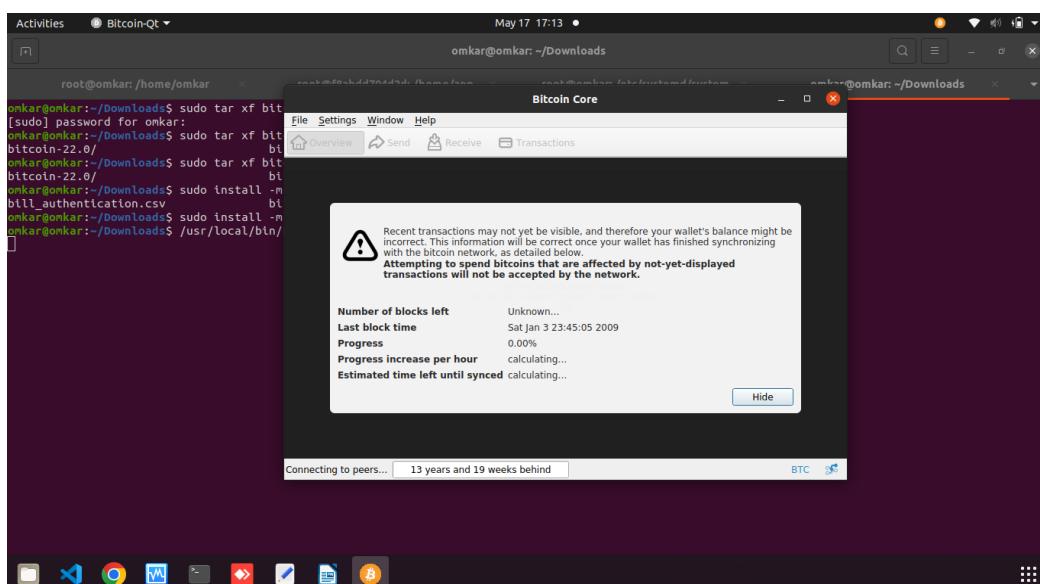
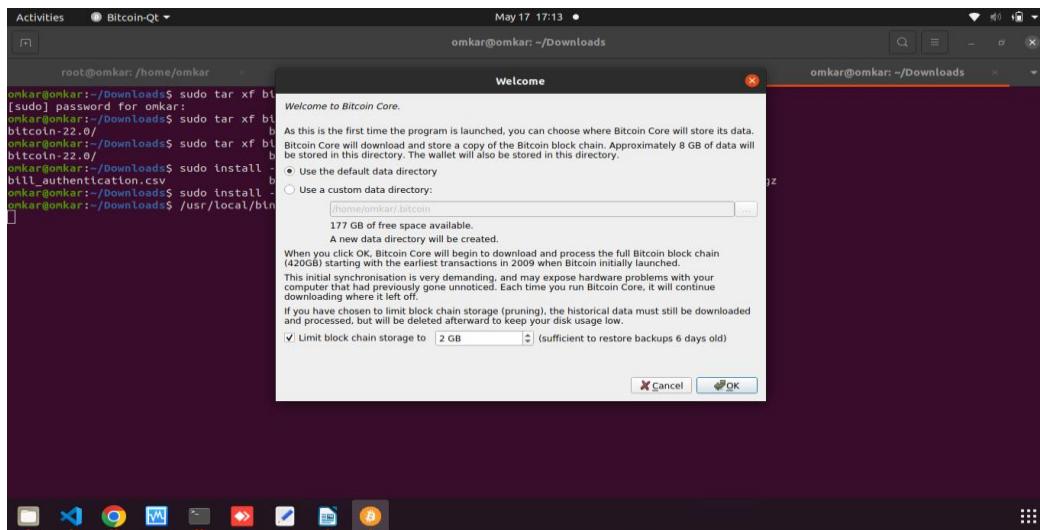
getmininginfo - Returns a json object containing mining-related information.

getwalletinfo - Returns an object containing various wallet state info.

listwallets - Returns a list of currently loaded wallets.



```
root@omkar:~/Downloads$ sudo tar xf bitcoin-22.0-x86_64-linux-gnu.tar.gz
[sudo] password for omkar:
root@omkar:~/Downloads$ sudo tar xf bitcoin-22.0
bitcoind-22.0/
root@omkar:~/Downloads$ sudo tar xf bitcoin-22.0
bitcoind-22.0/
root@omkar:~/Downloads$ sudo install -m 0755 -o root -g root -t /usr/local/bin bitcoin-22.0/bin/*
bitll_authentication.csv
root@omkar:~/Downloads$ sudo install -m 0755 -o root -g root -t /usr/local/bin bitcoin-22.0/bin/*
root@omkar:~/Downloads$ /usr/local/bin/bitcoin-qt
```



The screenshot shows two windows of the Bitcoin-QT application. The top window is the 'Node window' which displays a JSON response from a wallet command. The bottom window is the 'Console' window showing a series of commands and their outputs.

Node window (Top Window):

```

stealing their wallet contents. Do not use this console without fully
understanding the ramifications of a command.
17:15:41 ⚡ Executing command using "omkar" wallet
17:15:41 ⚡ getbalance
17:15:41 ⚡ 0.00000000
17:16:01 ⚡ listwalletdir
17:16:01 ⚡ {
  "wallets": [
    {
      "name": "omkar"
    }
  ]
}
17:16:06 ⚡ listwallets
17:16:06 ⚡ [
  "omkar"
]

```

Console Window (Bottom Window):

```

17:16:27 ⚡ getblock
17:16:27 ⚡ getblock "blockhash" { verbosity }

If verbosity is 0, returns a string that is serialized, hex-encoded data for block 'hash'.
If verbosity is 1, returns an Object with information about block <hash>.
If verbosity is 2, returns an Object with information about block <hash> and information about each transaction.

Arguments:
1. blockhash (string, required) The block hash
2. verbosity (numeric, optional, default=1) 0 for hex-encoded data, 1 for a json object, and 2 for json object with transaction data

Result (for verbosity = 0):
"hex" (string) A string that is serialized, hex-encoded data for block 'hash'

Result (for verbosity = 1):
{
  "hash" : "hex",           (string) The block hash (same as provided)
  "confirmations" : n,      (numeric) The number of confirmations, or -1 if the block is not on the main chain
  "size" : n,                (numeric) The block size
  "strippedsize" : n,       (numeric) The block size excluding witness data
  "weight" : n,              (numeric) The block weight as defined in BIP 141
  "height" : n,              (numeric) The block height or index
  "version" : n,             (numeric) The block version
  "versionHex" : "hex",      (string) The block version formatted in hexadecimal
  "merkleroot" : "hex",      (string) The merkle root
  "tx" : [                  (json array) The transaction ids
    "hex",                  (string) The transaction id
    ...
  ],
  "time" : xxx,              (numeric) The block time expressed in UNIX epoch time
  "mediantime" : xxx,        (numeric) The median block time expressed in UNIX epoch time
  "nonce" : n,                (numeric) The nonce
  "bits" : "hex",            (string) The bits
  ...
}

```

PRACTICAL 9

Aim: Create your own blockchain and demonstrate its use.

Steps & Output:

2. Open Postman and execute the requests in the given order as per screenshots.

http://127.0.0.1:5000/mine_block

GET http://127.0.0.1:5000/mine_block

Status: 200 OK Time: 11ms Size: 347B

```
* *Index*: 2
* *PreviousHash*:康纳的链，你真好！
* *previous_hash*: "ea829f38d6581133fe756ee80d5cb3f0e0039c97e6d38ea02d72e1c0baef99e",
* *proof*: 531,
* *timestamp*: "2021-09-18 13:13:18.998479"
```

http://127.0.0.1:5000/get_chain

GET http://127.0.0.1:5000/get_chain

Status: 200 OK Time: 9ms Size: 400B

```
* *chain*: [
*   {
*     *index*: 0,
*     *previous_hash*: "康纳的链，你真好！",
*     *merit*: 1,
*     *timestamp*: "2021-09-18 13:13:18.998479"
*   },
*   {
*     *index*: 1,
*     *previous_hash*: "ea829f38d6581133fe756ee80d5cb3f0e0039c97e6d38ea02d72e1c0baef99e",
*     *merit*: 1,
*     *timestamp*: "2021-09-18 13:13:18.998479"
*   }
* ],
* *length*: 2
```

The screenshot shows the Postman application interface. At the top, there are tabs for Home, Workspaces, Reports, and Explore. Below the tabs, there's a search bar labeled "Search Postman". The main workspace shows an overview with three active requests: "http://127.0.0.1:5000/is_valid" (status 200 OK), "http://127.0.0.1:5000/miner" (status 200 OK), and "http://127.0.0.1:5000/" (status 200 OK). The "http://127.0.0.1:5000/is_valid" request is expanded, showing a GET method to "http://127.0.0.1:5000/is_valid". The "Body" tab is selected, displaying the response body in JSON format:

```

1 | {
2 |   "message": "All good. The Blockchain is valid."
3 |
4 |
5 }

```

The status bar at the bottom indicates "Status 200 OK Time 5ms Size 104 B Save Response".

The screenshot shows the Postman application interface. At the top, there are tabs for Home, Workspaces, Reports, and Explore. Below the tabs, there's a search bar labeled "Search Postman". The main workspace shows an overview with three active requests: "http://127.0.0.1:5000/is_valid" (status 200 OK), "http://127.0.0.1:5000/miner" (status 200 OK), and "http://127.0.0.1:5000/" (status 200 OK). The "http://127.0.0.1:5000/miner" request is expanded, showing a GET method to "http://127.0.0.1:5000/miner". The "Body" tab is selected, displaying the response body in JSON format:

```

1 | {
2 |   "index": 3,
3 |   "message": "Congratulations, you just mined a block!",
4 |   "previous_hash": "62f4d19954be1878a4c37bd51a0f7ex381e522737471495-566495661820a32",
5 |   "proof": 41231,
6 |   "timestamp": "2021-09-16 11:34:10.210032"
7 |
8 }

```

The status bar at the bottom indicates "Status 200 OK Time 8ms Size 308 B Save Response".

The image contains two vertically stacked screenshots of the Postman application interface.

Screenshot 1: This screenshot shows a GET request to `http://127.0.0.1:5000/get_chain`. The response body is a JSON object representing a blockchain chain:

```

1 {
2   "chain": [
3     {
4       "index": 1,
5       "previous_hash": "0",
6       "proof": "333",
7       "timestamp": "2021-05-18 11:12:19.493468"
8     },
9     {
10       "index": 2,
11       "previous_hash": "eab2091e06058113de7756ed8f5e37da003c97bd3bae5d72e1d50dc99a",
12       "proof": "4553",
13       "timestamp": "2021-05-18 11:12:19.994791"
14     },
15     {
16       "index": 3,
17       "previous_hash": "629b3199546e05794e9776052e807fe303c322737477497c51640906082fb02",
18       "proof": "4553",
19       "timestamp": "2021-05-18 11:14:18.253832"
20     }
21   ],
22   "length": 3
23 }

```

Screenshot 2: This screenshot shows a POST request to `http://127.0.0.1:5000/is_valid`. The response body is a JSON object with a single key-value pair:

```

1 {
2   "message": "All good. The Blockchain is valid."
3 }

```

3. While executing the above steps in Postman, Python console will also track the requests

The image shows a terminal window displaying the log output of a Flask application named "blockchain". The log includes several requests from the IP address 127.0.0.1:

```

* Serving Flask app "blockchain" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [16/May/2021 11:13:16] "GET /mine_block
HTTP/1.1" 200 -
127.0.0.1 - - [16/May/2021 11:13:45] "GET /get_chain
HTTP/1.1" 200 -
127.0.0.1 - - [16/May/2021 11:13:57] "GET /is_valid
HTTP/1.1" 200 -
127.0.0.1 - - [16/May/2021 11:14:10] "GET /mine_block
HTTP/1.1" 200 -
127.0.0.1 - - [16/May/2021 11:14:16] "GET /get_chain
HTTP/1.1" 200 -
127.0.0.1 - - [16/May/2021 11:14:21] "GET /is_valid
HTTP/1.1" 200 -

```

CODE:

```
#Module 1 - Create a blockchainimport datetime
```

```
import hashlib
```

```
import json
```

```
from flask import Flask,jsonify
```

```
#Each block needs to have it's own timestamp- datetime#For hashing functions- hashlib
```

```
#Encode blocks in json before hashing- json#Web UI for block chain – flask
```

```
#return the messages to postman when we interact with our blockchain.
```

```
#1- Building our blockchain class Blockchain:
```

```
def __init__(self):
```

```
    self.chain=[]
```

```
    self.create_block(proof=1, previous_hash='0')
```

```
    #Call this future function to create block .
```

Takes two arguments - proof because each block of the block chain will have its own proof, previous hash.

```
def create_block(self, proof, previous_hash):
```

```
    block = {'index': len(self.chain) + 1, 'timestamp': str(datetime.datetime.now()),
```

```
            'proof': proof, 'previous_hash': previous_hash}
```

```
    self.chain.append(block)
```

```
    return block
```

```
    #proof is kinda like Nonce I guess
```

```
def get_previous_block(self):
```

```
    return self.chain[-1]
```

```
def proof_of_work(self, previous_proof):
```

```
    new_proof = 1
```

```
check_proof = False

while check_proof is False:

    hash_operation = hashlib.sha256(str(new_proof**2 - previous_proof**2).encode()).hexdigest()

    if hash_operation[:4] == '0000':

        check_proof = True

    else:

        new_proof += 1

return

new_proof

def hash(self, block):

    stringified_block = json.dumps(block, sort_keys = True).encode()

    # use json.dumps to ocnvert dict into string of json format for easy export into json return
    hashlib.sha256(stringified_block).hexdigest()

def is_chain_valid(self, chain):

    previous_block = chain[0]

    block_index = 1

    while block_index < len(chain):

        block = chain[block_index]

        if block['previous_hash'] != self.hash(previous_block):

            return False

        previous_proof = previous_block['proof']

        proof = block['proof']

        hash_operation = hashlib.sha256(str(proof**2 - previous_proof**2).encode()).hexdigest()

        if

            hash_operation[:4] != '0000':
```

```
return  
False  
  
previous_block = block block_index += 1  
  
return True  
  
# Part 2 - Mining our Blockchain  
  
# Creating a Web  
  
App app = Flask(__name__) app.config['JSONIFY_PRETTYPRINT_REGULAR'] = False  
  
# Creating a Blockchain  
  
object blockchain = Blockchain()  
  
# Creating a Web  
  
App app = Flask(__name__)  
  
app.config['JSONIFY_PRETTYPRINT_REGULAR'] = False  
  
# Creating a Blockchain  
  
object blockchain = Blockchain()  
  
# Mining a new block @app.route('/mine_block', methods = ['GET'])  
  
def mine_block():  
  
    previous_block = blockchain.get_previous_block()  
  
    previous_proof = previous_block['proof']  
  
    proof = blockchain.proof_of_work(previous_proof)  
  
    previous_hash = blockchain.hash(previous_block)  
  
    block = blockchain.create_block(proof, previous_hash)  
  
    response = {'message': 'Congratulations, you just mined a block!', 'index': block['index'],  
    'timestamp': block['timestamp'], 'proof': block['proof'], 'previous_hash':  
    block['previous_hash']} return jsonify(response), 200  
  
# Getting the full Blockchain
```

```
@app.route('/get_chain', methods = ['GET']) def get_chain():

    response = {'chain': blockchain.chain, 'length': len(blockchain.chain)}

    return jsonify(response), 200

# Checking if the Blockchain is valid

@app.route('/is_valid', methods = ['GET'])

def is_valid():

    is_valid = blockchain.is_chain_valid(blockchain.chain)

    if is_valid:

        response = {'message': 'All good. The Blockchain is valid.'}

    else:

        response = {'message': 'Houston, we have a problem. The Blockchain is not valid.'}

    return jsonify(response), 200 # Running the app app.run(host = '0.0.0.0', port = 5000)

#0.0.0.0 will make server available publicly
```

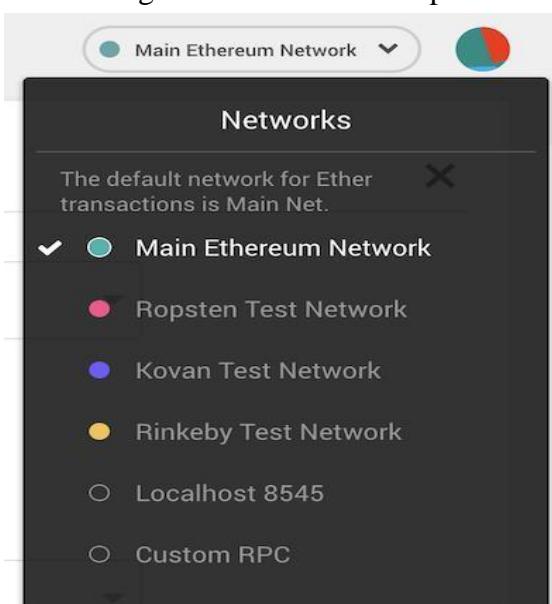
Connect to MetaMask

Let's back up for a second. As you'll recall, you started a network via ganache-cli on port 8545.

> ganache-cli -p 8545

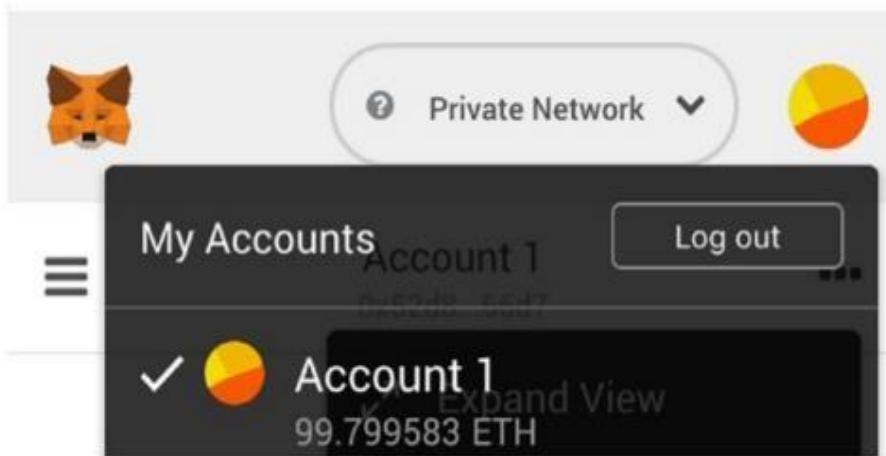
And you connected Truffle to the network.

> truffle migrate --network development



Then you were able to connect on port 8545 and run commands in Terminal. You can now connect MetaMask in a browser. To connect, select MetaMask and select Localhost 8545 in the drop-down menu

```
> ganache-cli -p 8545 -m 'journey badge medal slender behind junk develop produce spy enemy transfer room'
```



```
> ganache-cli -p 8545
```

After logging out, the welcome screen comes back with a link under it that says, —Import using account seed phrase

Create Password

New Password (min 8 chars)

Confirm Password

CREATE

Import with seed phrase



Now you can paste the mnemonic by selecting a password and clicking Restore

[< Back](#)

Import an Account with Seed Phrase

Enter your secret twelve word phrase here to restore your vault.

Wallet Seed

```
candy maple cake sugar pudding cream honey rich smooth  
crumble sweet treat
```

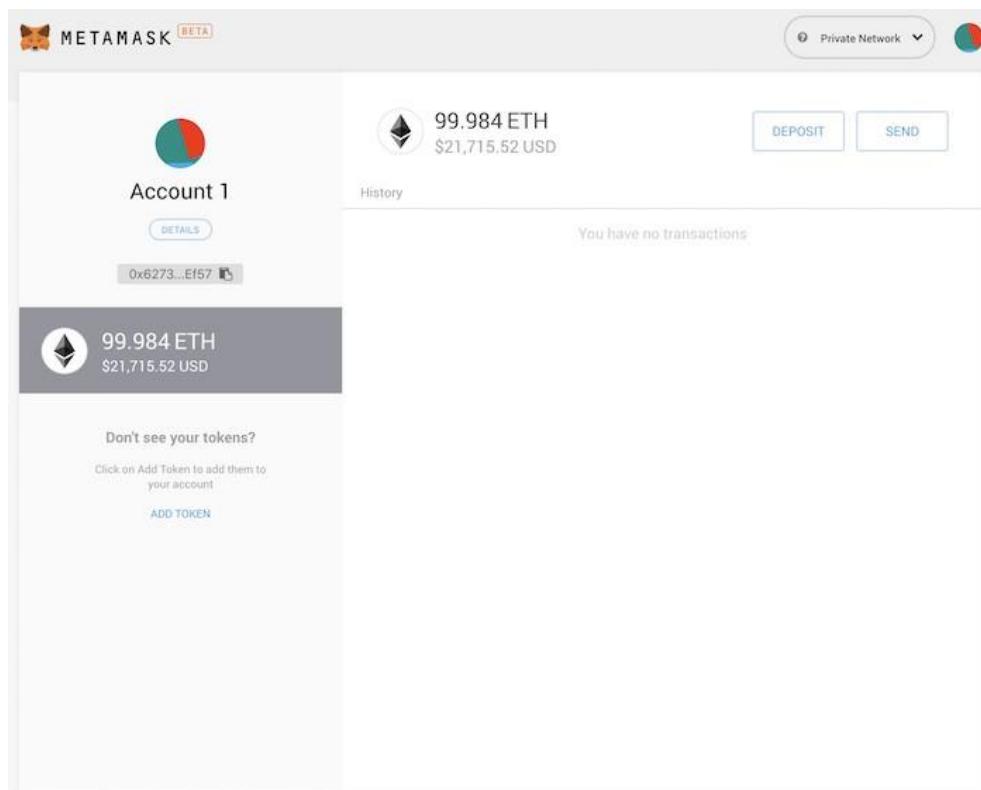
New Password (min 8 chars)

.....

Confirm Password

.....

IMPORT



Test Your Dapp Functionality

Next, fill in the form and initialize a transfer. Notice that MetaMask opens to confirm the transfer. This is an extra measurement of security to ensure only authorized transfers get approved.



PRACTICAL 10

Aim: Build Apps with angular.

Angular is a [TypeScript-based open-source framework](#), whose main purpose is developing web applications. But where Angular really shines is the creation of client applications and it is regarded as one of the best tools when it comes to single-page apps development as well. Angular community consists of more than 18 million users and that number is simply impressive.

A big part in this is played by tools and setup that Angular possesses: RxJS, Angular CLI, and different [code editors](#) that support the framework. RxJS is a reactive programming library that is pretty crucial when it comes to working with Angular. RxJS' main aim is to handle asynchronous data with multiple events and by doing that this reactive programming library allows engineers to build multiple channels of data exchange to ease the consumption of resources. So, basically, RxJS is similar to a conveyor for [JavaScript](#) codes, as it allows parallel and continuing execution of events in a manner, independent from one another, and without waiting for one event to happen to complete another. And although RxJS' learning curve might seem a bit high, it is worth every penny.

The second great tool in Angular's arsenal is the Angular command-line interface (or just CLI for short). Favored by many an engineer, Angular CLI is easy to set up, quite understandable even for newcomers, is packed to the brim with different testing tools right out of the box and its commands can be described as nothing but simple.

Angular Pros

1. Angular architecture is component-based and its primary architectural characteristic is the basis of components hierarchy. This fact allows developers to achieve a higher code quality by making the overall code more accessible and understandable by encapsulating all of the components with their functionality.

All Angular components are reusable. And this advantage is a direct outcome of the previous one because the previously mentioned encapsulation of components makes them

exceptionally self-sufficient. It also allows developers to reuse them in different parts of their applications, making the process of developing a bit faster and more convenient.

1. Angular's readability is off the [charts](#). Once again due to the component-based architecture and the encapsulation. Thus, new developers, albeit new to the whole [app](#) developing a game or just new to the project, can read code in a better way and reach their plateau of productivity quicker.

Angular is unit-test friendly. Try and guess why it is so. Right you are, all because of the component-based structure that simplifies the [quality assurance procedures](#) even when it comes to the smallest parts of the app, which are, of course, units.

Angular uses [TypeScript](#). Let us get a little misconception out of the way first – it is not mandatory to use TypeScript with Angular, as it provides devs with options on how to use their libraries more efficiently, including Redux and Flux. But why use them if you can use TypeScript, which can be described as a superset for JavaScript? Yes, it has its fair share of things to nitpick and yes, you basically have to learn another language if you never worked with TypeScript, but its overall usefulness is immense. Especially if you work on an enterprise-level project, as TypeScript simply has better navigation, autocompletion, refactoring services and it helps you to spot and get rid of common mistakes while you type in the code. All in all, TypeScript is great, and Angular is only better because of it.

Angular Cons

The main disadvantage of Angular is the fact that Angular is complex. Although the component-based architecture Angular possesses is great, the way in which components are managed is not, as each and every component in your app will, most likely, need dependency injections and all of them will definitely need lifecycle interfaces. And that's not mentioning the fact that you will have to rely on third-party libraries that are quite specific when it comes to Angular. Thus, developing apps on Angular can be (bear in mind that it is a possibility and not an axiom) pretty repetitive and tiresome.

Summing up, even though it can be fearsome for new developers and at times complicated

for new and professional developers alike, Angular is a great tool for a variety of web [app development](#) needs and, with Google's Long-Term Support, its future in the industry looks as bright as ever. That being said, let's get to the metaphorical cherry on top of the cake of today's article – an example of step-by-step Angular app development.

Step-by-step Angular App Creation

1. Install Angular CLI 8;
2. Proceed with Angular 8 Project creation;
3. Add Angular HttpClient;
4. Create UI Component;
5. Routing addition;
6. Build UI with Angular [Material](#) Components;
7. Set up a REST API mocking;
8. Use Angular HttpClient in order to consume the REST API;
9. HTTP Errors Handling is step number nine;
10. [Pagination](#) Addition;
11. Angular Application Firebase building and deployment.