

INDEX

Sr.no	Title	Pg.No	Date	Sign
1	Write a simple client class that generates the private and public keys by using the built-in Python RSA algorithm and test it.			
2	Setting up Ethereum network by using Geth command line interface.(INSTALLATION)			
3	Transfer ethers from one contract to another on an Ethereum testnet			
4	Transfer ethers from one account to another on an Ethereum testnet.			
5	Implement and demonstrate the use of the following in Solidity:			
5a	Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs, Mappings, Conversions, Ether Units, Special Variables			
5b	Functions, Function Modifiers, View functions, Pure Functions, Fallback Function, Function Overloading, Mathematical functions, Cryptographic functions			
6	Implement and demonstrate the use of the following in Solidity.			
6a	Withdrawal Pattern, Restricted Access			
6b	Contracts, inheritance, Constructors, Abstract Contracts, Interfaces			
6c	Libraries, Assembly, Events, Error handling.			
7	Deploying contracts on an external blockchain by using Ganache and/or MyEtherwallet, Metamask			
8	Deploy a local private blockchain over a network with Ethereum or Rust (VM)			
9	Implement the mining module of Bitcoin client . The mining module, or miner, should produce blocks that solve proof-of-work puzzle			
10	Compile and test smart contracts on a testing framework using the Ethereum Virtual Machine (EVM).			
11	Demonstrate the running of the blockchain node			
12	Demonstrate the use of Bitcoin Core API.			
13	Create your own blockchain and demonstrate its use			

PRACTICAL 1

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

Code:-

```
import hashlib
import random
import binascii
import datetime
import collections
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Cipher 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')

Gayatri = Client()
print ("sender ",Gayatri.identity)
```

Output:-

```
import hashlib
import random
import binascii
import datetime
import collections
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Cipher 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')

Gayatri = Client()
print ("sender ",Gayatri.identity)

sender 30819f300d06092a864886f70d0101050003818d0030818902818100cb786eb7f9418677fcdb7474d9444fdbae51c172ee890f562e0d06e75b7054f28b06adbfa5a62b52a5d94
a203e063ff48633edba87a3bafb5716f3892d483d58650647ff8861fc58680782f1f9a428c391c5b77b824ce2a484215371bf4567b53170a6f7f98010e067ad88cfda1650e63555da88984
8ab8284737863e370203010001
```

PRACTICAL 2

Aim:Setting up Ethereum network by using Geth command line interface.(INSTALLATION)

Aim: Setting up Ethereum network by using Geth command line interface.(INSTALLATION)

Install on Ubuntu via PPAs

The easiest way to install go-ethereum on Ubuntu-based distributions is with the built-in launchpad PPAs (Personal Package Archives). We provide a single PPA repository that contains both our stable and development releases for Ubuntu versions trusty, xenial, zesty and artful.

Code:-

linux:

To enable our launchpad repository run:

Step 1: open new terminal

Step 2: on terminal type this command

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

#if above command gives error then run

```
#sudo apt-get install --reinstall ca-certificates
```

Step 3: install the stable version of go-ethereum:

```
sudo apt-get update
```

```
sudo apt-get install ethereum
```

linux:

To enable our launchpad repository run:

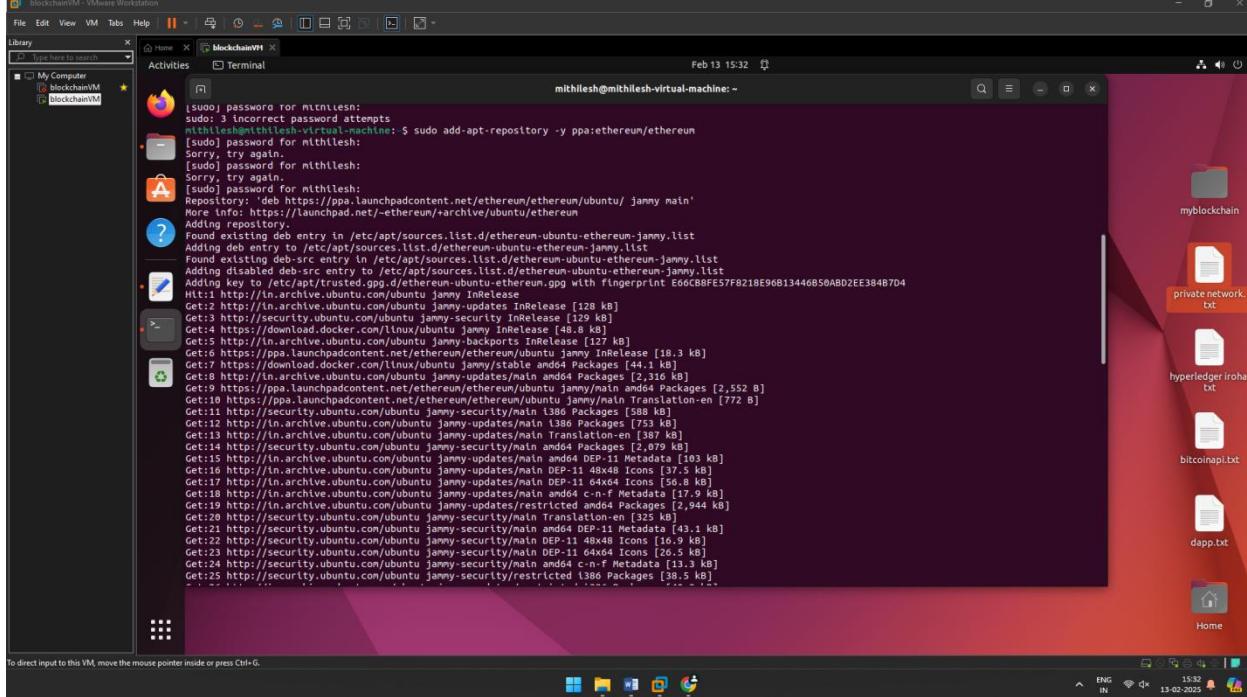
Step 1: open new terminal

Step 2: on terminal type this command

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

#if above command gives error then run

```
#sudo apt-get install --reinstall ca-certificates
```



```
mithilesh@mithilesh-virtual-machine: ~
lsudo password for mithilesh:
sudo: 3 incorrect password attempts
mithilesh@mithilesh-virtual-machine: ~$ sudo add-apt-repository -y ppa:ethereum/ethereum
[sudo] password for mithilesh:
Sorry, try again.
[sudo] password for mithilesh:
Sorry, try again.
[sudo] password for mithilesh:
Repository: 'deb https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu/ jammy main'
More info: https://launchpad.net/~ethereum/+archive/ubuntu/ethereum
Adding repository.
Found existing deb entry in /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Adding deb entry to /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Found existing deb-src entry in /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ethereum-ubuntu-ethereum.gpg with fingerprint E66CB8FE57F8218E96B13446B50ABD2EE384B7D4
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:6 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy InRelease [18.3 kB]
Get:7 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [44.1 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,316 kB]
Get:9 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy/main amd64 Packages [2,552 B]
Get:10 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy/main Translation-en [772 B]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [588 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [753 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [387 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2,079 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [103 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 48x48 Icons [37.5 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 64x64 Icons [56.8 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.9 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2,944 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [325 kB]
Get:21 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.1 kB]
Get:22 http://security.ubuntu.com/ubuntu jammy-security/main DEP-11 48x48 Icons [16.9 kB]
Get:23 http://security.ubuntu.com/ubuntu jammy-security/main DEP-11 64x64 Icons [26.5 kB]
Get:24 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.3 kB]
Get:25 http://security.ubuntu.com/ubuntu jammy-security/restricted i386 Packages [38.5 kB]
...
mithilesh@mithilesh-virtual-machine: ~
found existing deb-src entry in /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ethereum-ubuntu-ethereum.gpg with fingerprint E66CB8FE57F8218E96B13446B50ABD2EE384B7D4
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:6 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy InRelease [18.3 kB]
Get:7 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [44.1 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,316 kB]
Get:9 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy/main amd64 Packages [2,552 B]
Get:10 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy/main Translation-en [772 B]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [588 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [753 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [387 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2,079 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [103 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 48x48 Icons [37.5 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 64x64 Icons [56.8 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.9 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2,944 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [325 kB]
Get:21 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.1 kB]
Get:22 http://security.ubuntu.com/ubuntu jammy-security/main DEP-11 48x48 Icons [16.9 kB]
Get:23 http://security.ubuntu.com/ubuntu jammy-security/main DEP-11 64x64 Icons [26.5 kB]
Get:24 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.3 kB]
Get:25 http://security.ubuntu.com/ubuntu jammy-security/restricted i386 Packages [38.5 kB]
Get:26 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted i386 Packages [40.3 kB]
Get:27 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2,839 kB]
Get:28 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [515 kB]
Get:29 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]
Get:30 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted DEP-11 48x48 Icons [29 B]
Get:31 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted DEP-11 64x64 Icons [29 B]
Get:32 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted DEP-11 64x64@2 Icons [29 B]
Get:33 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [612 B]
Get:34 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,187 kB]
Get:35 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [757 kB]
Get:36 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [291 kB]
Get:37 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [359 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [497 kB]
```

```
mithilesh@mithilesh-virtual-machine: ~
Get:34 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,187 kB]
Get:35 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [757 kB]
Get:36 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [291 kB]
Get:37 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [359 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [497 kB]
Get:39 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 48x48 Icons [250 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/restricted DEP-11 48x48 Icons [29 B]
Get:42 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 64x64 Icons [402 kB]
Get:43 http://security.ubuntu.com/ubuntu jammy-security/restricted DEP-11 64x64 Icons [29 B]
Get:44 http://security.ubuntu.com/ubuntu jammy-security/restricted DEP-11 64x64@2 Icons [29 B]
Get:45 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [580 B]
Get:46 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [26.4 kB]
Get:47 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse i386 Packages [4,752 B]
Get:48 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [44.5 kB]
Get:49 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [11.5 kB]
Get:50 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:51 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [440 B]
Get:52 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.7 kB]
Get:53 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [961 kB]
Get:54 http://in.archive.ubuntu.com/ubuntu jammy-backports/main i386 Packages [59.9 kB]
Get:55 http://in.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.1 kB]
Get:56 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [7,056 B]
Get:57 http://in.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 48x48 Icons [9,524 B]
Get:58 http://in.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 64x64 Icons [11.2 kB]
Get:59 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:60 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Get:61 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted DEP-11 48x48 Icons [29 B]
Get:62 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted DEP-11 64x64 Icons [29 B]
Get:63 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted DEP-11 64x64@2 Icons [29 B]
Get:64 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [30.0 kB]
Get:65 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe i386 Packages [18.4 kB]
Get:66 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.6 kB]
Get:67 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.8 kB]
Get:68 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 48x48 Icons [19.7 kB]
Get:69 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 64x64 Icons [28.2 kB]
Get:70 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [672 B]
Get:71 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Get:72 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse DEP-11 48x48 Icons [29 B]
Get:73 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse DEP-11 64x64 Icons [29 B]
Get:74 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse DEP-11 64x64@2 Icons [29 B]

mithilesh@mithilesh-virtual-machine: ~
Get:53 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [961 kB]
Get:54 http://in.archive.ubuntu.com/ubuntu jammy-backports/main i386 Packages [59.9 kB]
Get:55 http://in.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.1 kB]
Get:56 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [7,056 B]
Get:57 http://in.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 48x48 Icons [9,524 B]
Get:58 http://in.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 64x64 Icons [11.2 kB]
Get:59 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:60 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Get:61 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted DEP-11 48x48 Icons [29 B]
Get:62 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted DEP-11 64x64 Icons [29 B]
Get:63 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted DEP-11 64x64@2 Icons [29 B]
Get:64 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [30.0 kB]
Get:65 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe i386 Packages [18.4 kB]
Get:66 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.6 kB]
Get:67 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.8 kB]
Get:68 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 48x48 Icons [19.7 kB]
Get:69 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 64x64 Icons [28.2 kB]
Get:70 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [672 B]
Get:71 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Get:72 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse DEP-11 48x48 Icons [29 B]
Get:73 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse DEP-11 64x64 Icons [29 B]
Get:74 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse DEP-11 64x64@2 Icons [29 B]
Get:75 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [650 kB]
Get:76 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [205 kB]
Get:77 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [125 kB]
Get:78 http://security.ubuntu.com/ubuntu jammy-security/universe DEP-11 48x48 Icons [82.0 kB]
Get:79 http://security.ubuntu.com/ubuntu jammy-security/universe DEP-11 64x64 Icons [122 kB]
Get:80 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [19.5 kB]
Get:81 http://security.ubuntu.com/ubuntu jammy-security/multiverse i386 Packages [1,356 kB]
Get:82 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.6 kB]
Get:83 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [8,260 B]
Get:84 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 DEP-11 Metadata [208 B]
Get:85 http://security.ubuntu.com/ubuntu jammy-security/multiverse DEP-11 48x48 Icons [29 B]
Get:86 http://security.ubuntu.com/ubuntu jammy-security/multiverse DEP-11 64x64 Icons [29 B]
Get:87 http://security.ubuntu.com/ubuntu jammy-security/multiverse DEP-11 64x64@2 Icons [29 B]
Get:88 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [224 B]
Fetched 20.0 MB in 16s (1,260 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
mithilesh@mithilesh-virtual-machine: ~
```

Step 3: install the stable version of go-ethereum:

sudo apt-get update

```
mithilesh@mithilesh-virtual-machine: ~
Get:87 http://security.ubuntu.com/ubuntu jammy-security/multiverse DEP-11 64x64@2 Icons [29 B]
Get:88 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [224 B]
Fetched 20.0 MB in 16s (1,260 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
mithilesh@mithilesh-virtual-machine: $ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Hit:6 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy InRelease
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [7,048 B]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [216 B]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.8 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Fetched 152 kB in 3s (55.8 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
```

sudo apt-get install ethereum

```
mithilesh@mithilesh-virtual-machine: $ sudo apt-get install ethereum
Reading package lists...
Building dependency tree...
Reading state information...
The following package was automatically installed and is no longer required:
  bootnode
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  ethereum
1 upgraded, 0 newly installed, 0 to remove and 536 not upgraded.
Need to get 1,454 B of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy amd64 ethereum amd64 1.15.0+build30732+jammy [1,454 B]
Fetched 1,454 B in 1s (1,669 B/s)
(Reading database ... 194923 files and directories currently installed.)
Preparing to unpack .../ethereum_1.15.0+build30732+jammy_amd64.deb ...
Unpacking ethereum (1.15.0+build30732+jammy) over (1.11.5+build28443+jammy) ...
Setting up ethereum (1.15.0+build30732+jammy) ...
```

```
mithilesh@mithilesh-virtual-machine: $ mkdir myblockchain2
mkdir: cannot create directory `myblockchain2': File exists
mithilesh@mithilesh-virtual-machine: $ cd myblockchain2
cd: command not found
mithilesh@mithilesh-virtual-machine: $ -cd myblockchain2
Command '-cd' not found, did you mean:
  command 'hcd' from deb hfsutils (3.2.6-15build1)
  command 'mcwd' from deb mtools (4.0.33-1+really4.0.32-1build1)
  command 'becd' from deb openssh-client (2.17-29)
Try: sudo apt install <deb name>
mithilesh@mithilesh-virtual-machine: $ cd myblockchain2
mithilesh@mithilesh-virtual-machine:~/myblockchain2$ geth account new --datadir data
INFO [02-15|18:06:01.146] Maximum peer count           ETH=50 LES=0 total=50
INFO [02-15|18:06:01.148] Smartcard socket not found, disabling   err="stat /run/pcscd/pcscd.comm: no such file or directory"
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:
Your new key was generated

Public address of the key: 0x8ab55c1f93cf6db6fc239e1e5ce6350e221d103
Path of the secret key file: data/keystore/UTC--2025-02-15T12:36:06.943056353Z--eb55c1f93cf6db6fc239e1ec5ce6350e221d103
- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!

mithilesh@mithilesh-virtual-machine:~/myblockchain2$ {
  "config": {
    "chainId": 12345,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "eip158BBlock": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "berlinBlock": 0,
    "ethash": {}
  },
  "difficulty": "1",
  "gasLimit": "8000000",
  "alloc": {
    "7df9a875a174b3bc565e6424a0050ebc1b2d1d02": { "balance": "300000" },
    "Efaf4df699211972a7D2C3306d1F778a1603F10F": { "balance": "400000" }
  }
}
```

To direct input to this VM, move the mouse pointer inside or press Ctrl-G.

blackchainVM - VMware Workstation

File Edit View VM Help

Library Type here to search

Activities Terminal

Feb 15 18:11

mithilesh@mithilesh-virtual-machine: ~/myblockchain2

```
        "ethash": {}  
    },  
    "difficulty": "1",  
    "gasLimit": "8000000",  
    "alloc": {}  
    "7df9ab75a174b3bc56e6424a0850ebc1b2d1d82": { "balance": "3000000" },  
    "Efaf4df869211972a7D2C3306d1f778a1003f10F": { "balance": "4000000" }  
}  
  
config:: command not found  
challenge:: command not found  
highestStedBlock:: command not found  
ep158Block:: command not found  
ep158Block:: command not found  
ep158Block:: command not found  
byzantiumBlock:: command not found  
constantBlock:: command not found  
petropolisBlock:: command not found  
istanbulBlock:: command not found  
berlinBlock:: command not found  
ethash:: command not found  
): command not found  
difficulty:: command not found  
gasLimits:: command not found  
alloc:: command not found  
7df9ab75a174b3bc56e6424a0850ebc1b2d1d82:: command not found  
Efaf4df869211972a7D2C3306d1f778a1003f10F:: command not found  
bash: syntax error near unexpected token `'  
[root@mithilesh ~]# cd /home/mithilesh/myblockchain2$ sudo nano genesis.json  
[sudo] password for mithilesh:  
mithilesh@mithilesh-virtual-machine:~/myblockchain2$ geth init --datadir data genesis.json  
INFO [02-15|18:07:08.693] Maximum peer count ETHE=50 LEST=0 total=50  
INFO [02-15|18:07:08.695] Smartcard socket not found, disabling err="start /pcscd/pcscd.comm: no such file or directory"  
INFO [02-15|18:07:08.711] Set global gas cap cap=50,000,000  
INFO [02-15|18:07:08.725] Using leveldb as the backing database database=/home/mithilesh/myblockchain2/data/geth/chaindata cache=16.00MiB handles=16  
INFO [02-15|18:07:08.725] Allocated cached file handles database=/home/mithilesh/myblockchain2/data/geth/chaindata/ancient/chain readonly=false  
INFO [02-15|18:07:08.748] Using LevelDB as the backing database INFO [02-15|18:07:08.771] Opened ancient database  
INFO [02-15|18:07:08.773] Writing custom genesis block databasesize=3 size=397.008 time=3.810527ms gcnodes=0 gcsizes=0.008 gctime=0s liveblocks=1 livesize=0.008  
INFO [02-15|18:07:08.773] Persisted trie from memory database databasesize=3 size=397.008 time=3.810527ms gcnodes=0 gcsizes=0.008 gctime=0s liveblocks=1 livesize=0.008  
INFO [02-15|18:07:08.779] Using leveldb as the backing database databasesize=3 size=397.008 time=3.810527ms gcnodes=0 gcsizes=0.008 gctime=0s liveblocks=1 livesize=0.008  
INFO [02-15|18:07:08.799] Allocated cache and file handles databasesize=3 size=397.008 time=3.810527ms gcnodes=0 gcsizes=0.008 gctime=0s liveblocks=1 livesize=0.008  
INFO [02-15|18:07:08.794] Using LevelDB as the backing database databasesize=3 size=397.008 time=3.810527ms gcnodes=0 gcsizes=0.008 gctime=0s liveblocks=1 livesize=0.008  
INFO [02-15|18:07:08.804] Opened ancient database databasesize=3 size=397.008 time=3.810527ms gcnodes=0 gcsizes=0.008 gctime=0s liveblocks=1 livesize=0.008  
INFO [02-15|18:07:08.804] Writing custom genesis block databasesize=3 size=397.008 time=637.684us gcnodes=0 gcsizes=0.008 gctime=0s liveblocks=1 livesize=0.008  
INFO [02-15|18:07:08.805] Persisted trie from memory database databasesize=3 size=397.008 time=637.684us gcnodes=0 gcsizes=0.008 gctime=0s liveblocks=1 livesize=0.008
```

```
blockchainVM - VMware Workstation
File Edit View VM Help || Home Activities Terminal Feb 15 18:12 ⓘ
Library Type here to search
[+] My Computer
[+] blockchainVM *
[+] blockchainVM10

mitchilesh@mitchilesh-virtual-machine: ~/myblockchain2
INFO [02-15] [18:07:19.826] Loaded most recent local block
INFO [02-15] [18:07:19.826] Failed to load snapshot
INFO [02-15] [18:07:19.826] Rebuilding state snapshot
INFO [02-15] [18:07:19.827] Regenerated local transaction journal
INFO [02-15] [18:07:19.829] Gasprice oracle is ignoring threshold set
INFO [02-15] [18:07:19.829] Error reading unclear shutdown markers
WARN [02-15] [18:07:19.830] Engine API started but chain not configured
INFO [02-15] [18:07:19.830] Starting peer-to-peer node
INFO [02-15] [18:07:19.832] Resuming state snapshot generation
INFO [02-15] [18:07:19.832] Generated state snapshot
INFO [02-15] [18:07:19.862] New local node record
INFO [02-15] [18:07:19.862] IBC endpoint opened
INFO [02-15] [18:07:19.865] IBC channel opened
INFO [02-15] [18:07:19.872] Started P2P networking
bdf18e16bb9035ee5c567f03d2a127.0.0.1:30933
INFO [02-15] [18:07:19.872] WebSocket enabled
INFO [02-15] [18:07:19.872] HTTP server started
INFO [02-15] [18:07:30.620] Looking for peers
INFO [02-15] [18:07:30.659] Looking for peers
INFO [02-15] [18:07:50.697] Looking for peers
INFO [02-15] [18:08:00.737] Looking for peers
INFO [02-15] [18:08:10.773] Looking for peers
INFO [02-15] [18:08:20.798] Looking for peers
INFO [02-15] [18:08:30.891] Looking for peers
INFO [02-15] [18:08:40.959] Looking for peers
INFO [02-15] [18:08:51.869] Looking for peers
INFO [02-15] [18:08:53.833] Looking for peers
INFO [02-15] [18:09:12.102] Looking for peers
INFO [02-15] [18:09:22.424] Looking for peers
INFO [02-15] [18:09:32.431] Looking for peers
INFO [02-15] [18:09:42.475] Looking for peers
INFO [02-15] [18:09:52.643] Looking for peers
INFO [02-15] [18:10:02.711] Looking for peers
INFO [02-15] [18:10:12.936] Looking for peers
INFO [02-15] [18:10:23.028] Looking for peers
INFO [02-15] [18:10:33.194] Looking for peers
INFO [02-15] [18:10:43.268] Looking for peers
INFO [02-15] [18:10:53.382] Looking for peers
INFO [02-15] [18:11:03.932] Looking for peers
INFO [02-15] [18:11:13.961] Looking for peers
INFO [02-15] [18:11:23.965] Looking for peers
INFO [02-15] [18:11:34.354] Looking for peers
INFO [02-15] [18:11:44.503] Looking for peers
INFO [02-15] [18:11:54.723] Looking for peers
number=0 hash=c9fa51...402a28 tde1 age=55y1mo4d
err="missing or corrupted snapshot"
transactions=0 accounts=0
threshold=2
err="leveldb: not found"
elapsed=4.900ms
accounts=2 slots=0 storage=82.008 dangling=0
elapsed=5.527ms
seq=1,739,623,839,842 id=3791943c38d73c ip=127.0.0.1 udp=30303 tcp=30303
url=/home/mitchilesh/myblockchain2/data/geth
path=/home/mitchilesh/myblockchain2/data/geth/jwtsecret
sel=(node)/20fd444415933305fd2cd7cf2e005f71eb9ba5b25e4accbb371f82be330aa0b606720e74c9cb990e8773ba47c9ebd
uris=wss://127.0.0.1:8551
endpoint=127.0.0.1:8551 auth=true prefix= cors=localhost vhosts=localhost
peercount=0 tried=64 static=0
peercount=1 tried=55 static=0
peercount=2 tried=50 static=0
peercount=0 tried=26 static=0
peercount=0 tried=89 static=0
peercount=0 tried=21 static=0
peercount=0 tried=60 static=0
peercount=2 tried=32 static=0
peercount=0 tried=47 static=0
peercount=0 tried=59 static=0
peercount=1 tried=33 static=0
peercount=0 tried=71 static=0
peercount=0 tried=35 static=0
peercount=0 tried=54 static=0
peercount=0 tried=69 static=0
peercount=0 tried=35 static=0
peercount=0 tried=33 static=0
peercount=1 tried=30 static=0
peercount=0 tried=72 static=0
peercount=0 tried=30 static=0
peercount=0 tried=61 static=0
peercount=0 tried=50 static=0
peercount=0 tried=65 static=0
peercount=0 tried=55 static=0
peercount=0 tried=91 static=0
peercount=0 tried=34 static=0
peercount=0 tried=38 static=0

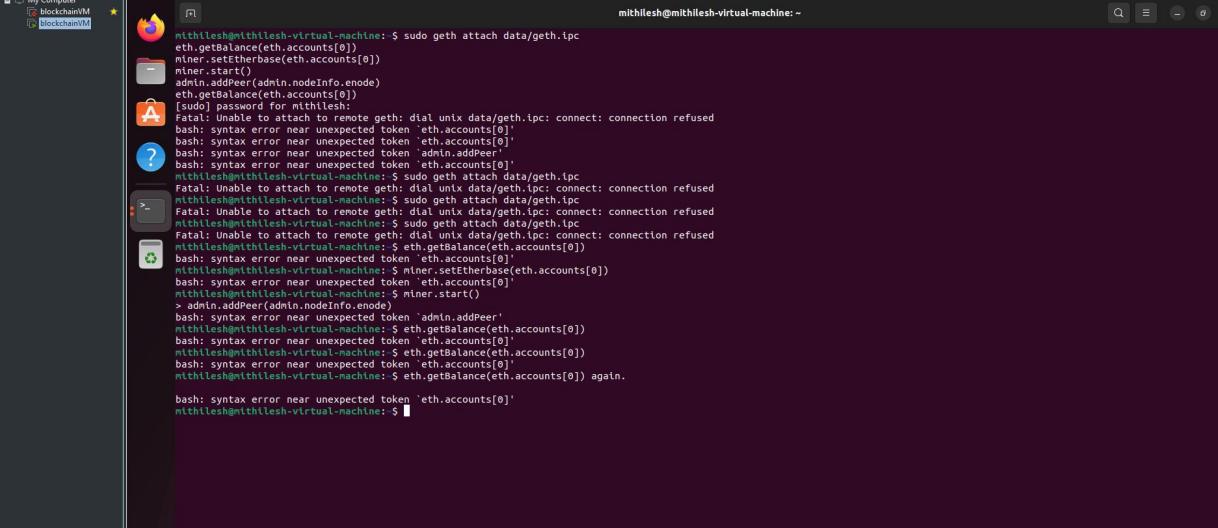
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

EN IN 18:12
15-02-2023
```

Terminal 2 :

Terminal 2 : **Output:**

Output:-



The screenshot shows a terminal window titled "BlockchainVM" running on a Linux desktop environment. The terminal displays a series of commands being run by a user named "mithilesh". The commands involve interacting with a Ethereum node using the "geth" command, specifically attaching to a data socket and performing various operations like getting balances and adding peers. The terminal output includes several "Fatal" errors indicating connection refused to the geth socket, as well as syntax errors from the shell. The desktop interface includes a dock at the bottom with icons for various applications like a browser, file manager, and terminal.

```
mithilesh@mithilesh-virtual-machine: ~
```

```
sudo geth attach data/geth.ipc
eth.getBalance(eth.accounts[0])
miner.setEtherbase(eth.accounts[0])
miner.start()
admin.addPeer(admin.nodeInfo.enode)
eth.getBalance(eth.accounts[0])
sudo password for mithilesh
Fatal: Unable to attach to remote geth: dial unix data/geth.ipc: connect: connection refused
bash: syntax error near unexpected token `eth.accounts[0]`
bash: syntax error near unexpected token `eth.accounts[0]`
bash: syntax error near unexpected token `admin.addPeer()`
bash: syntax error near unexpected token `eth.accounts[0]`
mithilesh@mithilesh-virtual-machine: ~ sudo geth attach data/geth.ipc
Fatal: Unable to attach to remote geth: dial unix data/geth.ipc: connect: connection refused
mithilesh@mithilesh-virtual-machine: ~ sudo geth attach data/geth.ipc
Fatal: Unable to attach to remote geth: dial unix data/geth.ipc: connect: connection refused
mithilesh@mithilesh-virtual-machine: ~ sudo geth attach data/geth.ipc
Fatal: Unable to attach to remote geth: dial unix data/geth.ipc: connect: connection refused
mithilesh@mithilesh-virtual-machine: ~ eth.getBalance(eth.accounts[0])
bash: syntax error near unexpected token `eth.accounts[0]`
mithilesh@mithilesh-virtual-machine: ~ miner.setEtherbase(eth.accounts[0])
bash: syntax error near unexpected token `eth.accounts[0]`
mithilesh@mithilesh-virtual-machine: ~ miner.start()
> admin.addPeer(admin.nodeInfo.enode)
bash: syntax error near unexpected token `admin.addPeer()`
mithilesh@mithilesh-virtual-machine: ~ eth.getBalance(eth.accounts[0])
bash: syntax error near unexpected token `eth.accounts[0]`
mithilesh@mithilesh-virtual-machine: ~ eth.getBalance(eth.accounts[0])
bash: syntax error near unexpected token `eth.accounts[0]`
mithilesh@mithilesh-virtual-machine: ~ eth.getBalance(eth.accounts[0])
bash: syntax error near unexpected token `eth.accounts[0] again.
mithilesh@mithilesh-virtual-machine: ~ eth.getBalance(eth.accounts[0]) again.

bash: syntax error near unexpected token `eth.accounts[0]` again.
mithilesh@mithilesh-virtual-machine: ~
```

PRACTICAL 3

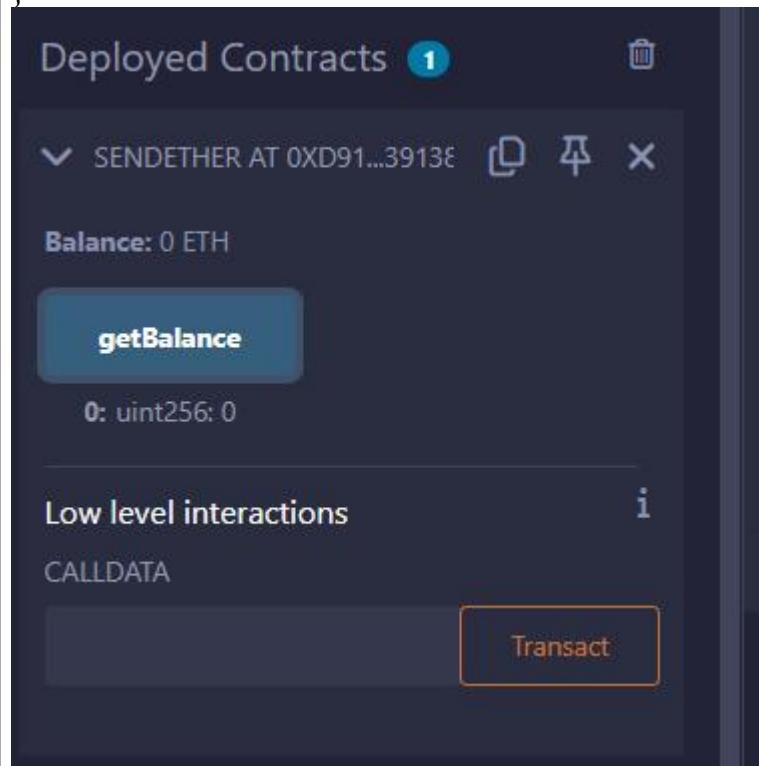
Aim:-

Code:-

Output:-

Transfer ethers from one **contract** to another on an Ethereum testnet.

```
pragma solidity ^0.8.0;
contract sendEther{
function getBalance() external view returns(uint)
{
    return address(this).balance;
}
receive() external payable { }
}
```



PRACTICAL 4

Aim:-

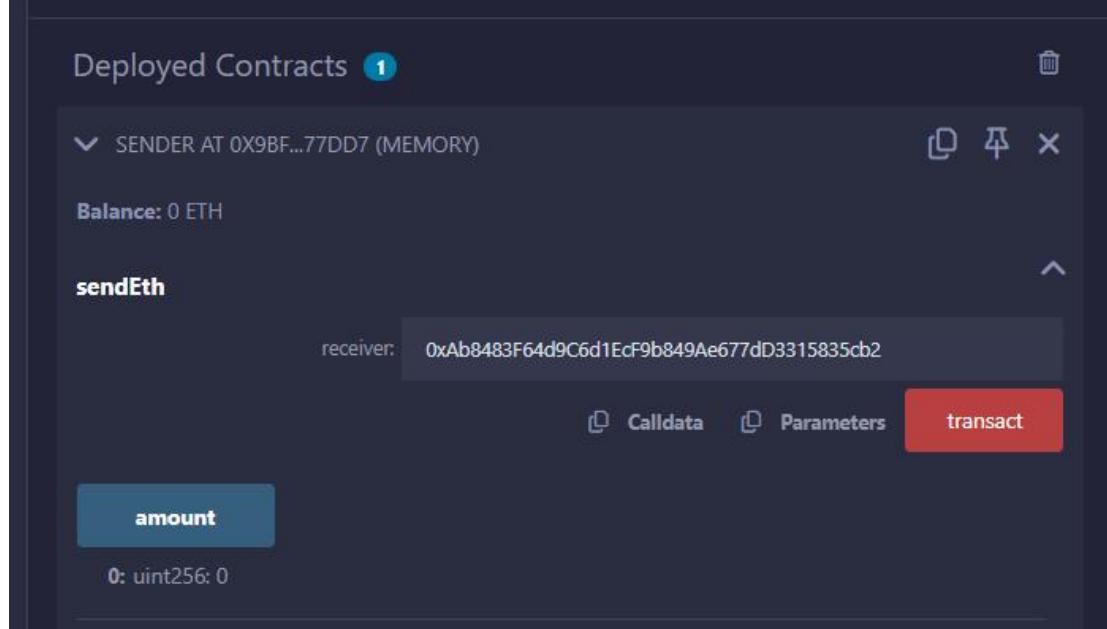
Code:-

Output: Transfer ethers from one **account** to another on an Ethereum testnet.

//<https://dev.to/sparklesix/solidity-tutorial-how-to-build-and-deploy-a-smart-contract-to-send-ether-from-one-account-to-another-n54>

```
pragma solidity ^0.8.11;
```

```
contract Sender {  
    uint public amount;  
    address payable owner;  
  
    constructor (){  
        owner = payable(msg.sender); // set the deployer of contract as the owner  
    }  
    function sendEth(address payable receiver) payable public{  
        require(owner == msg.sender, "Only the owner can send funds");  
        amount = msg.value;  
        receiver.transfer(amount);  
    }  
}
```



Transactions recorded 4 i >

Deployed Contracts 1

✓ SENDER AT 0XD7A...F771B (M) D P X

Balance: 0 ETH

sendEth

0xAb8...35cb2

amount

Low level interactions i

CALldata

10

Transact



[vm]

in Alert

Deployed Contracts 1

✓ SENDER AT 0XD7A...F771B (M) D P X

Balance: 0 ETH

sendEth

0xAb8...35cb2

amount

0: uint256: 0

```
0xAb8...35cb2 (99.99999999999824852 ether)
```

```
0x5B3...eddC4 (99.99999999999130854 ether)
```

```
0xAb8...35cb2 (99.99999999999824852 ether)
```

```
0x4B2...C02db (100 ether)
```

Ganach

<https://abhibvp003.medium.com/how-to-install-and-execute-truffle-on-an-ubuntu-16-04-7d0ff6458c9b>

<https://ethereum.stackexchange.com/questions/93533/call-an-existing-contract-function-from-truffle-console>

```
sudo apt-get -y install curl git vim build-essential  
sudo apt-get install curl software-properties-common
```

```
sudo apt install npm  
sudo npm install -g web3  
sudo apt-get install nodejs  
sudo apt install python3.9  
curl -sL https://deb.nodesource.com/setup_10.x | sudo bash -  
sudo npm install --global node-sass@latest  
sudo npm install -g truffle@latest  
sudo npm install -g ganache-cli  
export NODE_OPTIONS=--openssl-legacy-provider
```

```
////to update npm//  
sudo npm cache clean -f  
sudo npm install -g n
```

```
sudo n latest
```

```
///////////
```

Start from here!!!

```
mkdir upg1  
cd upg1  
truffle init
```

```
///////// create contract  
nano contracts/Helloworld.sol  
pragma solidity ^0.5.0;  
contract HelloWorld {  
    function sayHello() public pure returns(string memory){  
        return("hello world");  
    }  
}
```

```
///////////create configuration  
nano migrations/1_initial_migration.js  
const Migrations = artifacts.require("HelloWorld");
```

```
module.exports = function (deployer) {  
    deployer.deploy(Migrations,"hello");  
};
```

```
//////////network configuration
nano truffle-config.js
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1",
      port: 8545,
      network_id: "*",
    }
  }
}
//////////start ganache-cli
```

ganache-cli

```
/////////
truffle migrate
```

```
truffle console
#replace contact address
contract = await HelloWorld.at('0x37354B83aadd35516c56f24b724228f29300be77')
a = await contract.sayHello()
a
```

1. Transfer ethers from one **contract** to another on an Ethereum testnet.

```
pragma solidity ^0.8.11;
contract sendEther{
  function getBalance() external view returns(uint)
  {
    return address(this).balance;
  }
  receive() external payable { }
}
```

Deployed Contracts 1

SENDETHER AT 0XF8E...9FBE8 ()

Balance: 0 ETH

getBalance

0: uint256: 0

Low level interactions i

2. Transfer ethers from one **account** to another on an Ethereum testnet.

```
pragma solidity ^0.8.11;

contract Sender {
  uint public amount;
```

address payable owner;

```
constructor () {
    owner = payable(msg.sender); // set the deployer of contract as the owner
}
function sendEth(address payable receiver) payable public {
    require(owner == msg.sender, "Only the owner can send funds");
    amount = msg.value;
    receiver.transfer(amount);
}
}
```

Deployed Contracts 1

SENDER AT 0xD7A...F771B (ME) ⚙️ ✎ ✕

Balance: 0 ETH

sendEth 5 ▾

amount

0: uint256: 0

PRACTICAL 5

5Implement and demonstrate the use of the following in Solidity:

PRACTICAL 5a

Aim:-Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs, Mappings, Conversions, Ether Units, Special Variables

Code:-

A)Variables:

supports three types of variables.

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

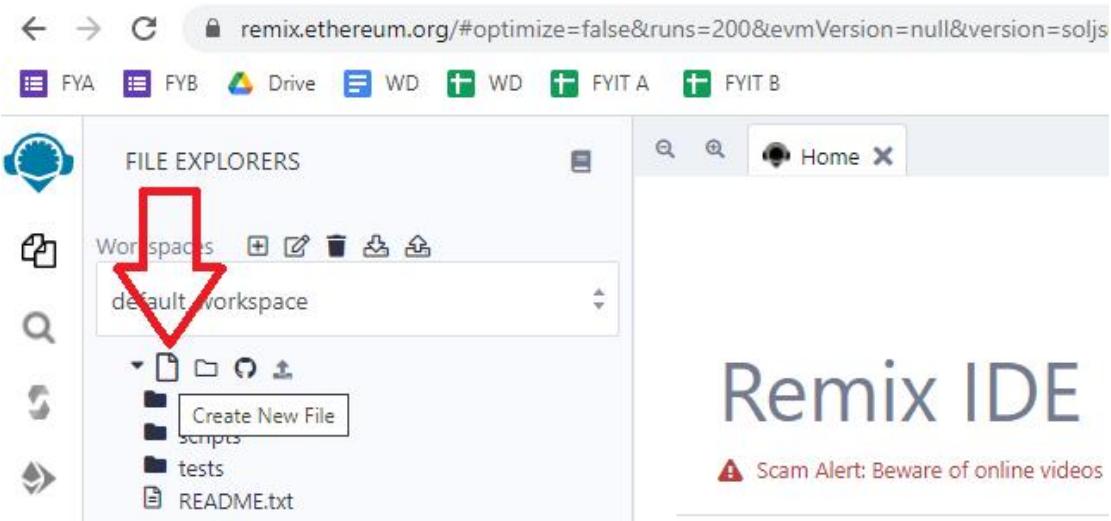
Local Variables – Variables whose values are present till function is executing.

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

Step 1: Open this website

<https://remix.ethereum.org/>

Step 2: Create new file – practical.sol



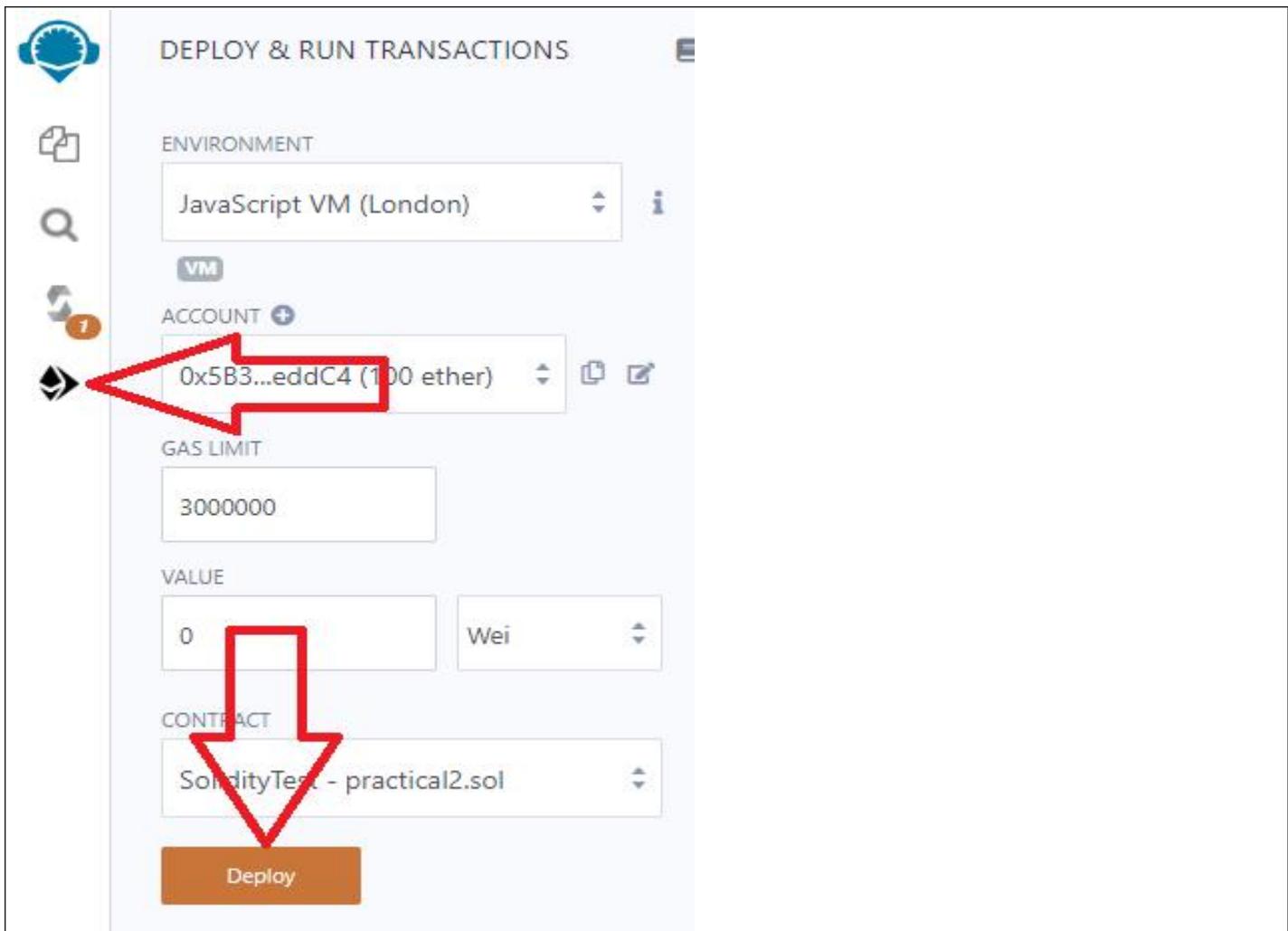
Step 3:Write the below program in new file

```
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
    }
}
```

Step 4: Compile contract

The screenshot shows the Solidity Compiler interface. On the left, there is a vertical toolbar with icons for file operations (New, Open, Save, Find, Copy, Paste, Undo, Redo). The main area is titled "SOLIDITY COMPILER". It includes a dropdown menu set to "default". Under "COMPILER CONFIGURATION", there are three checkboxes: "Auto compile" (unchecked), "Enable optimization" (unchecked), and "Hide warnings" (unchecked). A dropdown menu next to "Enable optimization" is set to "200". To the right of these settings is a button labeled "Estimated". Below the configuration is a large blue button with a circular arrow icon and the text "Compile practical2.sol". Underneath this button is a grey button labeled "Compile and run script" with an info icon (i) and a copy icon (c). At the bottom, there is a "CONTRACT" section with a dropdown menu currently showing "SolidityTest (practical2.sol)". A red arrow points from the "SolidityTest" dropdown down to the "Compile practical2.sol" button.

Step 5: Deploy contract



Step 6: Select the contract and click button

DEPLOY & RUN TRANSACTIONS

CONTRACT

SolidityTest - practical2.sol

Deploy

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 1

Deployed Contracts

SOLIDITYTEST AT 0xD91...39138 (MEM)

getResult

0: uint256: 3

```

1 pragma solidity ^0.5.0;
2 contract Soli...
3     uint stored...
4     constructor...
5     storedData = ...
6 }
7 function get...
8     uint a = 1;
9     uint b = 2;
10    uint result;
11    return resu...
12 }
13 }
14

```

Deployed Contracts 1

SOLIDITYTEST AT 0X7EF...8CB4:

Balance: 0 ETH

getResult

getResult - call

0: uint256: 3

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;
}
}
```

Transactions recorded 1 i >

Deployed Contracts 1 

SOLIDITY_VAR_TEST AT 0xD91.   

Balance: 0 ETH

state_var

0: uint8: 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;
}
}
```

Deployed Contracts 1 

SOLIDITY_VAR_TEST AT 0xD2A.   

Balance: 0 ETH

getResult

0: uint256: 3

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;
}
}

```

The screenshot shows the 'Deployed Contracts' section of the Truffle UI. It lists a single contract named 'TEST AT 0XDDA...5482D (MEMC)'. Below the contract name, it shows the balance as '0 ETH'. Under the 'admin' state variable, its address is listed as '0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4'. There are icons for deleting the contract and switching between tabs.

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;
}
```

▼ SOLIDITYTEST AT 0XB27...07C2! ⌂ ⌄ ✕

Balance: 0 ETH

a

0: uint16: 21

b

0: uint16: 9

dec

0: uint256: 9

diff

0: uint256: 10

div

0: uint256: 2

inc

0: uint256: 21

mod

0: uint256: 0

mul

0: uint256: 200

sum

0: uint256: 30

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;
}
```

Deployed Contracts 1

SOLIDITYTEST AT 0XCD6...99DF ⚙️ ⚡ ✎

Balance: 0 ETH

a

0: uint16: 20

b

0: uint16: 10

eq

0: bool: false

gtr

0: bool: true

gtreq

0: bool: true

les

0: bool: false

leseq

0: bool: false

noteq

0: bool: true

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);
}
}

```

Deployed Contracts 1

LOGICALOPERATOR AT 0X93F...

Balance: 0 ETH

Logic

a: 1

b: 0

Calldata Parameters **call**

0: bool: false
1: bool: true
2: bool: false

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 ;  
  
}
```

DEPLOY & RUN TRANSACTIONS

✓ > □

Deployed Contracts 1



▼ SOLIDITYTEST AT 0X5FD...9D88 ⌂ ⚡ ×

Balance: 0 ETH

a

0: uint16: 20

and

0: uint16: 0

b

0: uint16: 10

leftshift

0: uint16: 20480

not

0: uint16: 65515

or

0: uint16: 30

rightshift

0: uint16: 0

xor

0: uint16: 30

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 ;
}
}
```

DEPLOY & RUN TRANSACTIONS



SOLIDITYTEST AT 0X7B9...B6AC

Balance: 0 ETH

getResult

assign_div

0: uint256: 50

assign_mod

0: uint256: 32

assign_mul

0: uint256: 10

assign_sub

0: uint256: 50

assignment

0: uint16: 20

assignment_add

0: uint256: 50

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;
}
}

```

Deployed Contracts 1

SOLIDITYTEST AT 0xE28...4157f

Balance: 0 ETH

sub

a: 2

b: 6

Calldata Parameters call

0: uint256: 4

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 Pract3{
function test(int s, int e) public view returns(int)
{
int i;
int sum=0;
i=s;
while(i<=e)
{
sum+=i; //sum=sum+i;
i++;
}
return sum;
}
```

The screenshot shows the Truffle UI interface. At the top, it says "Deployed Contracts" with a count of 1. Below that, it shows a deployed contract named "PRACT3 AT 0X1C9...2B4BD (ME)". Underneath, it displays the "Balance: 0 ETH". A transaction is being prepared to call the "test" function. The "Parameters" section shows "s: 5" and "e: 7". At the bottom, there are buttons for "Calldata" and "Parameters", and a prominent blue "call" button. The result of the call is shown as "0: int256: 18".

2. Do-while loop:

```
pragma solidity ^0.5.0;
contract Pract3{
function test(int s, int e) public view returns(int)
{
int i;
int sum=0;
i=s;
do
{
sum+=i; //sum=sum+i;
i++;
}
```

```
}while(j<=e);  
return sum;  
}  
}
```

Deployed Contracts 1

PRACT3 AT 0X5A8...C4D01 (ME) ⚡ 平 X

Balance: 0 ETH

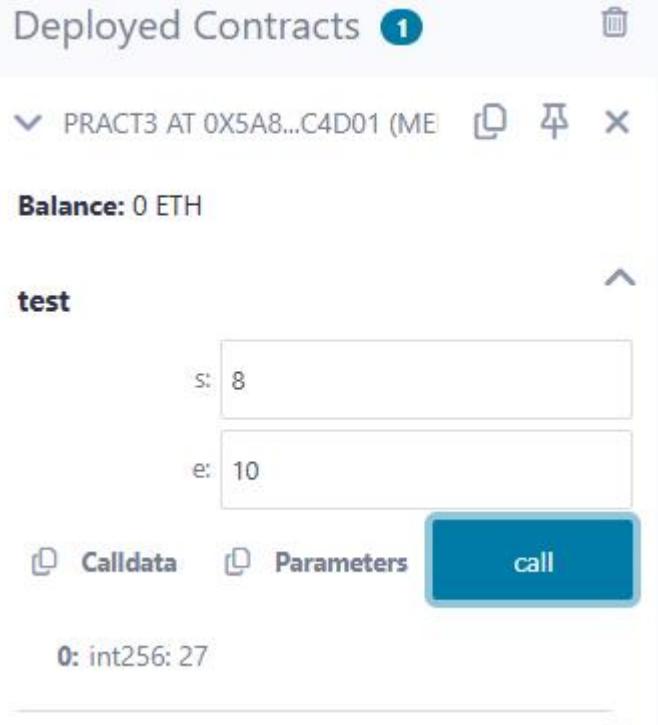
test

s: 8

e: 10

Calldata Parameters call

0: int256: 27



3. For Loop:

```
contract Pract3{  
function test(int s, int e) public view returns(int)  
{  
int i;  
int sum=0;  
for(i=s;i<=e;i++)  
{  
sum+=i; //sum=sum+i;  
}  
return sum;  
}
```

Deployed Contracts 1

PRACT3 AT 0X406...2CFBC (MEN) ⌂ ⌓ X

Balance: 0 ETH

test

S: 4

e: 8

Calldata Parameters call

0: int256: 30

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);  
}  
}  
(continue statement)
```

Deployed Contracts 1

PRACT3 AT 0X049...A1FD3 (MEI)

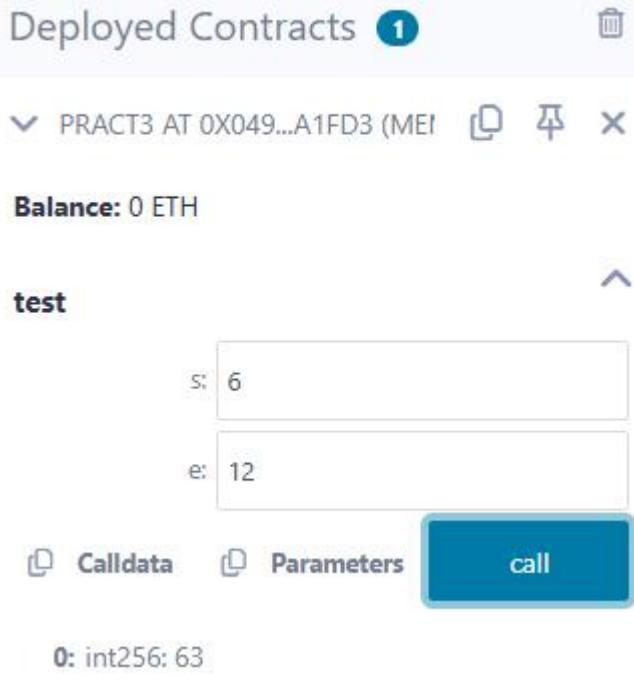
Balance: 0 ETH

test

s: 6
e: 12

Calldata Parameters **call**

0: int256: 63



```
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);
}
}

```



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
}}

```

Transactions recorded 31 i >

Deployed Contracts 1

SOLIDITYTEST AT 0xC3B...AAEC

Balance: 0 ETH

getResult

0: string: 3

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;
}
}

```

The screenshot shows the Solidity code for the `isEven` function. Below the code, there are three sections: **Local Variables**, **Return Value**, and **Global Variables**.

- Local Variables:** Shows the value of `even` as `0x00000000000000000000000000000000`.
- Return Value:** Shows the value of `return even;` as `0: Object`.
- Global Variables:** Shows the values of various blockchain variables:
 - `block.chainid: 3333`
 - `block.coinbase: 0x00000000000000000000000000000000`
 - `block.difficulty: 0`
 - `block.gaslimit: 52565`
 - `block.number: 2`
 - `block.timestamp: 1739806040`
 - `msg.sender: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4`
 - `msg.sig: 0x8873af24`
 - `msg.value: 0 Wei`
 - `tx.origin: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4`
 - `block.basefee: 1 Wei (1)`

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;  
}  
}
```

The screenshot displays the Truffle UI interface, which includes several panels:

- Top Left Panel:** Shows the contract code for `Types - 5.sol`. The code contains a function `decision_making` that returns a string based on the value of `i` (ranging from 0 to 10). It includes comments for readability.
- Top Right Panel:** The **DEPLOY & RUN TRANSACTIONS** panel shows the deployment of the contract. It includes fields for **CONTRACT** (`Types - 5.sol`), **evm version** (`istanbul`), and a **Deploy** button. Below it, there are buttons for **At Address** and **Load contract from Address**.
- Middle Left Panel:** The **Solidity Locals** panel indicates "No data available".
- Middle Center Panel:** The **Solidity State** panel shows the variable `i: 12 uint256` and the result `string`. The **Step details** panel provides a detailed trace of the current VM step, including gas usage and memory state.
- Middle Right Panel:** The **Transactions recorded** panel lists a pending transaction to `Types.decision_making` with address `0x9D7f74d0C41E726EC95884E0e97Fa6129e3b5E99`.
- Bottom Left Panel:** The **Storage [Completely Loaded]** panel shows storage slots `0x20` and `0x40` containing hex values.
- Bottom Center Panel:** The **Return Value** panel shows the return value as an object.
- Bottom Right Panel:** The **Global Variables** panel lists various global variables with their current values.

```
// 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

CONTRACT

Test - 1.sol

evm version: byzantium

Deploy

1

▼

 Publish to IPFS**At Address**

Load contract from Address

Transactions recorded **1**  ▼

-
- Run transactions using the latest compilation result

Save**Run**Deployed Contracts **1**  TEST AT 0xD91...39138 (MEMO)   **Balance:** 0 ETH**str_out**

0: string: 1

PRACTICAL 5a

Aim:- WRITE A SOLIDITY PROGRAM FOR STRING, ARRAYS, ENUMS, STRUCTURE & MAPPINGS.

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)

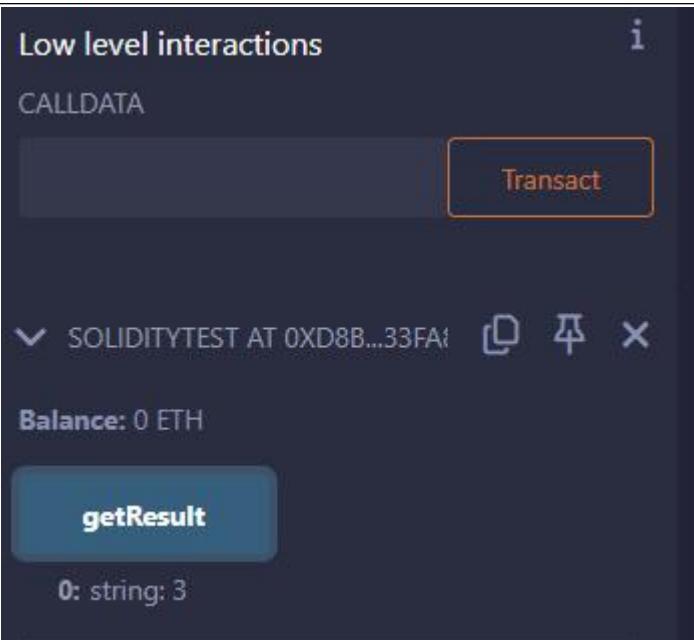
```
pragma solidity ^0.5.0;

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);
}
}
```



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;
function
// 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;
}
```



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
```

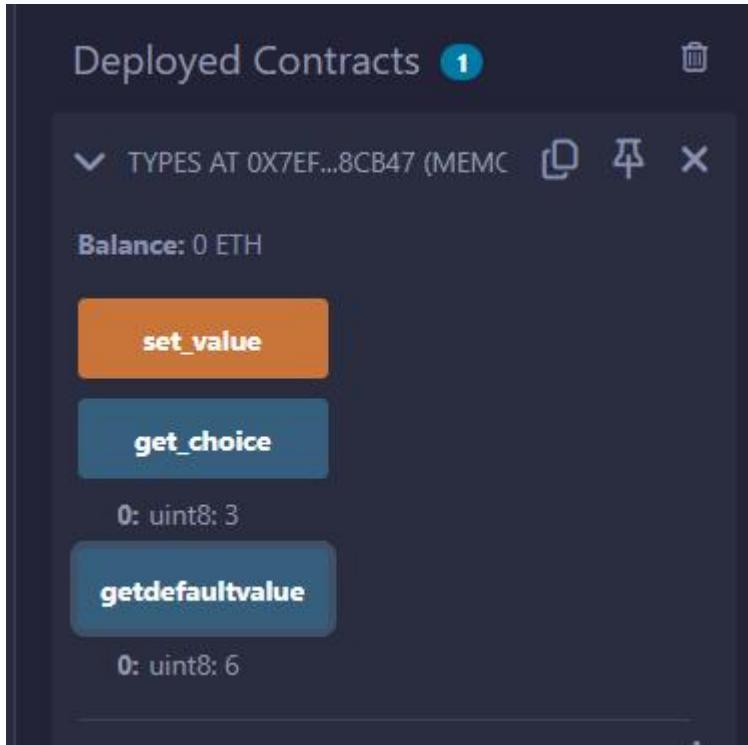
```
function get_choice(
```

```
) public view returns (week_days) {
```

```
return choice;
```

```
}
```

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



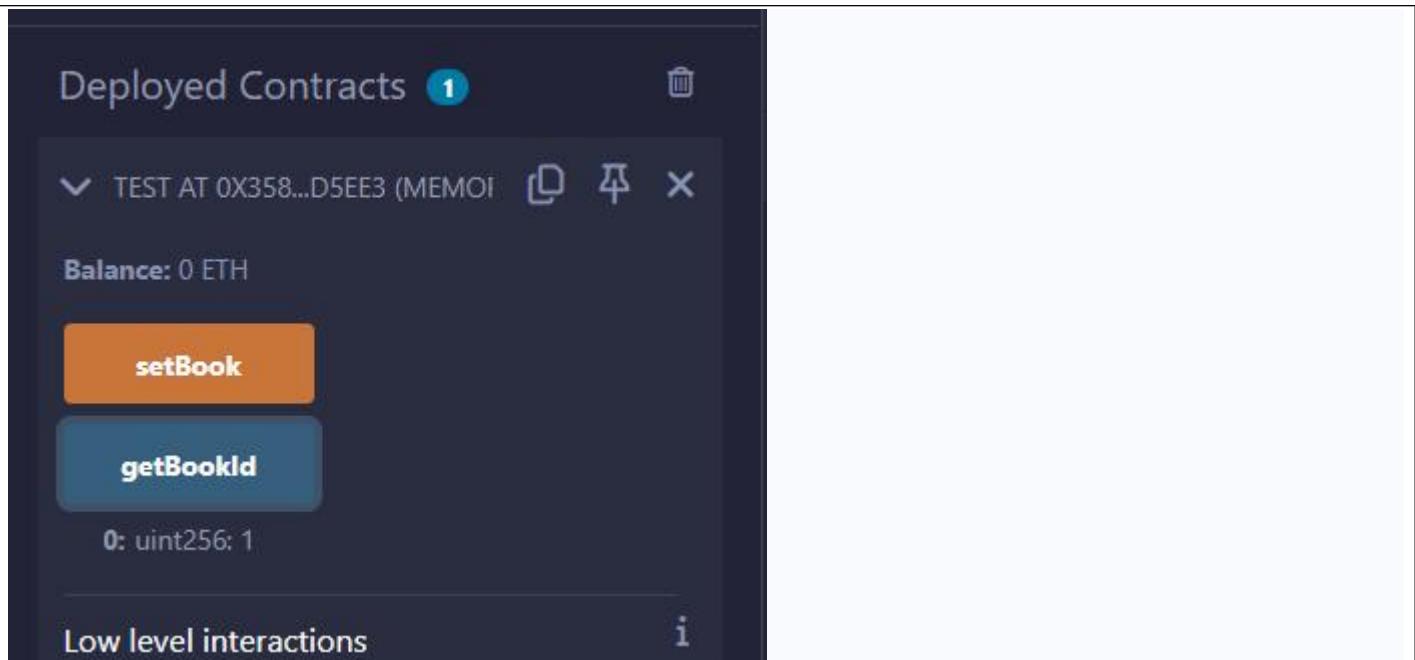
D)Structure:

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;
}
}
```



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];
    }
}
```

ENVIRONMENT ⚙️

Remix VM (Cancun) ⚙️

VM

ACCOUNT + 🖊️ 🗑️

0x4B2...C02db (99.9999999999...)

GAS LIMIT

Estimated Gas

Custom 3000000

VALUE

0 Wei ⚙️

CONTRACT

LedgerBalance - 1.sol ⚙️

Deployed Contracts 2

▼ LEDGERBALANCE AT 0X0FC...9A

Balance: 0 ETH

updateBalance

Low level interactions

CALldata

Transact

▼ LEDGERBALANCE AT 0X4B2...CC

Balance: 99.999999999999797472 ETH

updateBalance

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;
}
}
```

Deployed Contracts 1

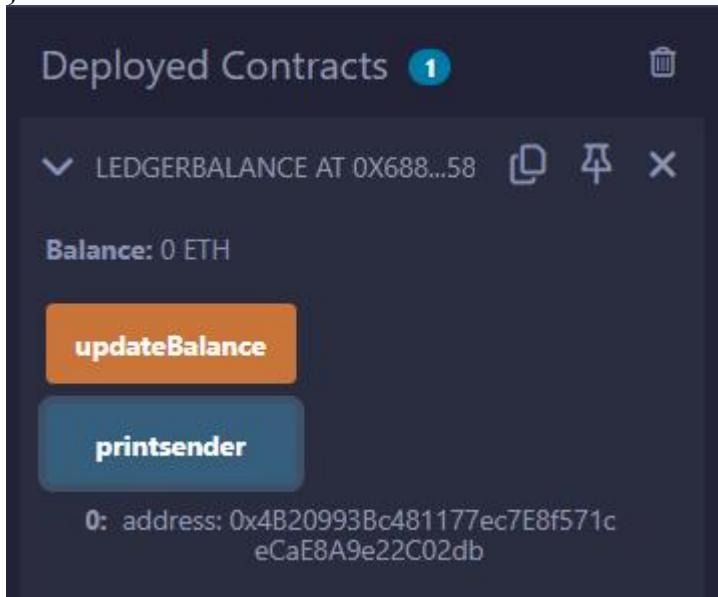
LEDGERBALANCE AT 0X688...58

Balance: 0 ETH

updateBalance

printsender

0: address: 0x4B20993Bc481177ec7E8f571c
eCaE8A9e22C02db



PRACTICAL 5B

Aim: WRITE A SOLIDITY PROGRAM FOR FUNCTION OVERLOADING, MATHEMATICAL FUNCTION & CRYPTOGRAPHIC FUNCTIONS.

Functions, Function Modifiers, View functions, Pure Functions, Fallback Function, Function Overloading, Mathematical functions, Cryptographic functions.

AIM: WRITE A SOLIDITY PROGRAM FOR FUNCTION, VIEW FUNCTION, PURE FUNCTION & FALBACK FUNCTION.

A)Function:

A function is a group of reusable code which can be called anywhere in your program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions.

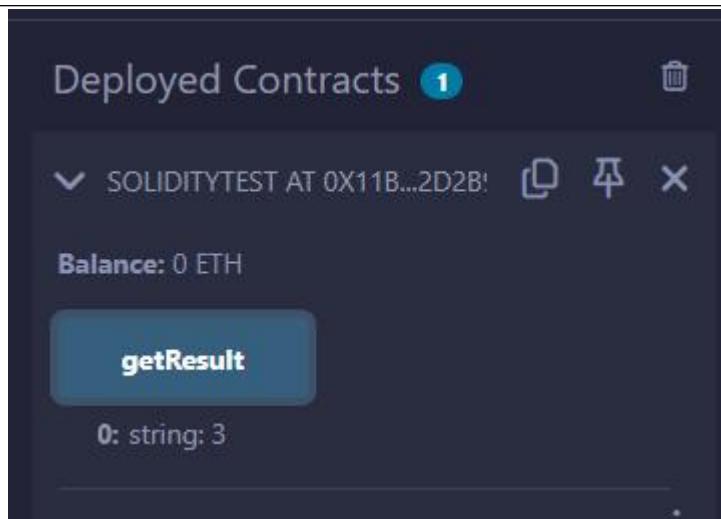
```
pragma solidity ^0.5.0;

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);//access local variable
}
}
```



B)View Function:

View functions ensure that they will not modify the state. A function can be declared as **view**. Getter method are by default view functions.

C)Pure Function:

Pure functions ensure that they not read or modify the state. A function can be declared as **pure**. Pure functions can use the revert() and require() functions to revert potential state changes if an error occurs.

```
pragma solidity ^0.5.0;
contract Test {
int public x=10; //global
int y=90;//state
function f1() public returns(int){
    //read and update is allowed
    x=100;
return x;
}
function f2() public view returns(int){
    // x=100; //erro beacuse x is global/state
    //we can access but we cannot update state or global variable int view function
return x;
}
function f3() public pure returns(int){
    //we cannot access or update state or global variable in pure function
    int z=80;
return z;
}
```

Deployed Contracts 1

SOLIDITYTEST AT 0xC4F...AD13

Balance: 0 ETH

getResult

0: string: 3

D)Fallback Function:

Fallback function is a special function available to a contract.

```
pragma solidity ^0.5.0;
contract Test {
uint public x ;
function() external { x = 1; }
}
contract Sink {
function() external payable { }
}
contract Caller {
function callTest(Test test) public returns (bool) {
(bool success,) = address(test).call(abi.encodeWithSignature("nonExistingFunction()"));
require(success);
// test.x is now 1
address payable testPayable = address(uint160(address(test)));
// Sending ether to Test contract,
// the transfer will fail, i.e. this returns false here.
return (testPayable.send(2 ether));
}
function callSink(Sink sink) public returns (bool) {
address payable sinkPayable = address(sink);
return (sinkPayable.send(2 ether));
}
}
```

Deployed Contracts 1

CALLER AT 0XA9D...6661D (MEN)

Balance: 0 ETH

callSink 1

callTest 1

creation of Caller pending...

[vm] from: 0x4B2...C02db to: Caller.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0xf93...dabb6

transact to Caller.callSink errored: Error encoding arguments: Error: invalid address (argument="address", value="1", code=INVALID_ARGUMENT, version=address/5.7.0) (argument=null, value=)

transact to Caller.callTest errored: Error encoding arguments: Error: invalid address (argument="address", value="1", code=INVALID_ARGUMENT, version=address/5.7.0) (argument=null, value=)

transact to Caller.callSink errored: Error encoding arguments: Error: invalid address (argument="address", value="1", code=INVALID_ARGUMENT, version=address/5.7.0) (argument=null, value=)

transact to Caller.callTest errored: Error encoding arguments: Error: invalid address (argument="address", value="1", code=INVALID_ARGUMENT, version=address/5.7.0) (argument=null, value=)

Deployed Contracts 1

CALLER AT 0xA9D...6661D (ME) Copy Edit X

Balance: 0 ETH

callSink

sink: "1"

Calldata Parameters transact

callTest

test: "1"

Calldata Parameters transact

PRACTICAL 6

Aim:-Implement and demonstrate the use of the following in Solidity.

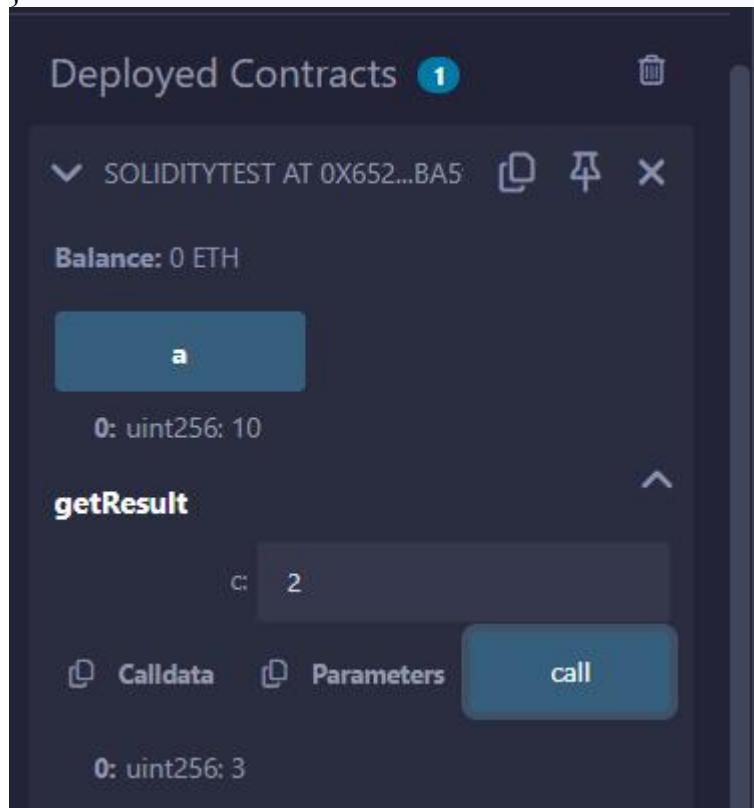
PRACTICAL 6a

6a. Withdrawal Pattern, Restricted Access.

Solidity

6a

```
pragma solidity ^0.5.0;
contract SolidityTest {
    uint storedData; // State variable
    uint public a=10;
    constructor() public {
        storedData = 10;
    }
    function getResult(uint c) public view returns(uint){
        uint a = 1; // local variable
        uint b = 2;
        uint result = a + b;
        return result; //access the state variable
    }
}
```



Function Overloading:

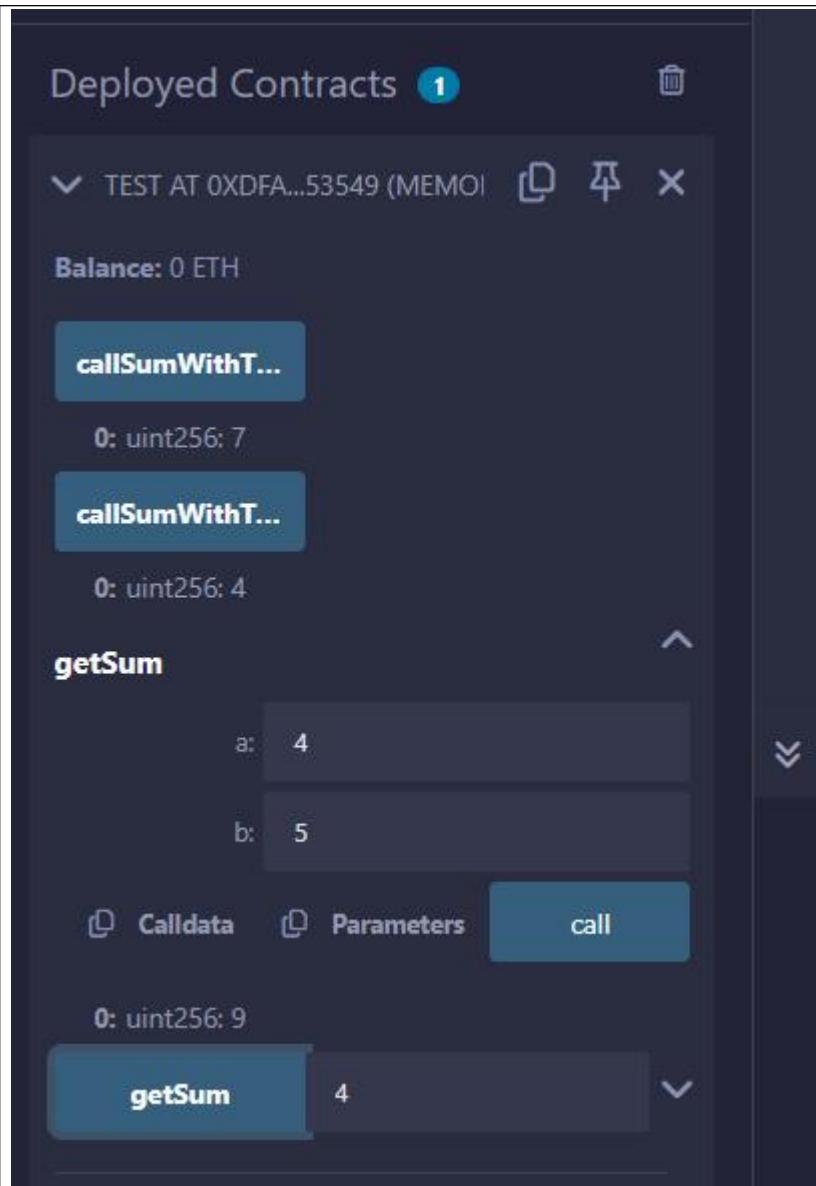
The definition of the function must differ from each other by the types and/or the number of arguments in the argument list. You cannot overload function declarations that differ only by return type.

```
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(2,2);  
}  
function callSumWithThreeArguments() public pure returns(uint){  
return getSum(1,2,4);  
}  
}
```





Mathematical Function:

Solidity provides inbuilt mathematical functions as well.

```
pragma solidity ^0.5.0;
```

```
contract Test {
function callAddMod() public pure returns(uint){
return addmod(4, 5, 3);
}
function callMulMod() public pure returns(uint){
return mulmod(4, 5, 3);
}
```

The screenshot shows a dark-themed interface for managing Ethereum contracts. At the top, it says "Deployed Contracts" with a count of "1". Below that, a single contract is listed: "TEST AT 0XD09...FCB49 (MEMO)". To the right of the contract name are three icons: a dropdown arrow, a copy symbol, and a delete symbol. Underneath the contract name, the text "Balance: 0 ETH" is displayed. Below this, there are two blue rectangular buttons labeled "callAddMod" and "callMulMod". Under each button, there is a small text output: "0: uint256: 0" under "callAddMod" and "0: uint256: 2" under "callMulMod".

Cryptographic Function:

Solidity provides inbuilt cryptographic functions as well.

```
pragma solidity ^0.5.0;
contract Test {
    function callKeccak256() public pure returns(bytes32 result){
        return keccak256("ABC");
    }
}
```

This screenshot shows a similar interface to the first one, but with a different deployed contract. The top bar says "Deployed Contracts" with a count of "1". A single contract is listed: "TEST AT 0X285...CE934 (MEMO)". To the right of the contract name are three icons: a dropdown arrow, a copy symbol, and a delete symbol. Below the contract name, the text "Balance: 0 ETH" is shown. There is a blue rectangular button labeled "callKeccak256". Underneath the button, the text "0: bytes32: result 0xe1629b9dda060bb30c7 908346f6af189c16773fa148d3366701fba a35d54f3c8" is displayed.

PRACTICAL 6B

Aim:- WRITE A SOLIDITY PROGRAM FOR CONTRACT, INHERITANCE, CONSTRUCTORS, ABSTRACT CONTRACTS, INTERFACES, LIBRARIES, ASSEMBLY, EVENTS, ERROR HANDLING.

A)Contract:

Contract in Solidity is similar to a Class in C++. A Contract have following properties.

Constructor – A special function declared with constructor keyword which will be executed once per contract and is invoked when a contract is created.

State Variables – Variables per Contract to store the state of the contract.

Functions – Functions per Contract which can modify the state variables to alter the state of a contract.

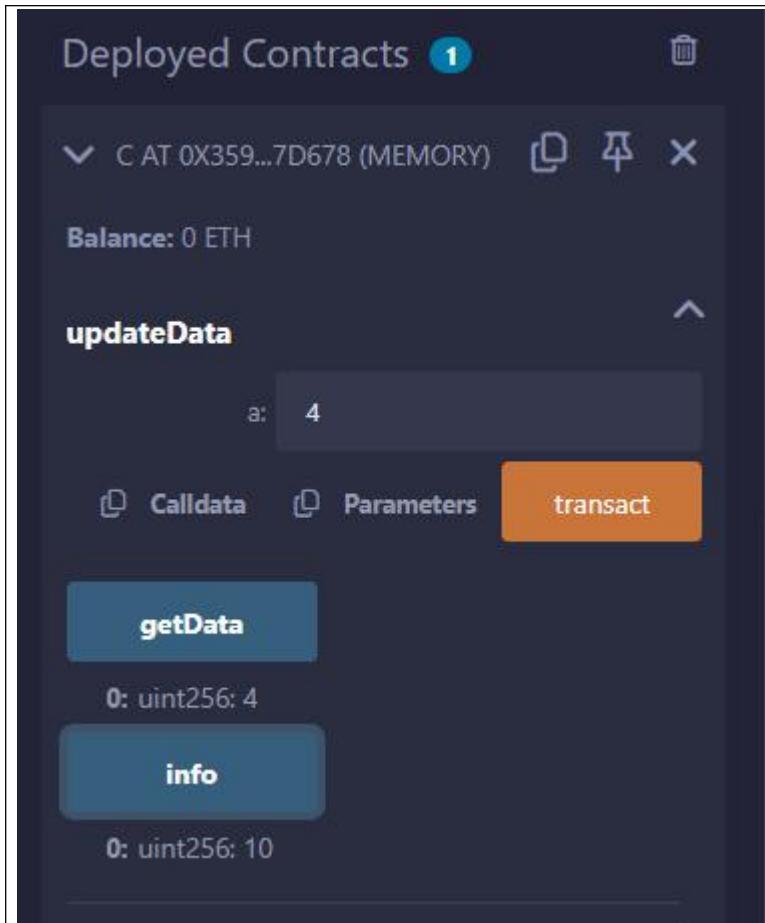
// Calling function from external contract

```
pragma solidity ^0.5.0;
contract C {
//private state variable
uint private data;

//public state variable
uint public info;

//constructor
constructor() public {
info = 10;
}
//private function
function increment(uint a) private pure returns(uint) { return a + 1; }
//public function
function updateData(uint a) public { data = a; }
function getData() public view returns(uint) { return data; }
function compute(uint a, uint b) internal pure returns (uint) { return a + b; }
}
//Derived Contract
contract E is C {
uint private result;
C private c;

constructor() public {
c = new C();
}
function getComputedResult() public {
result = compute(3, 5);
}
function getResult() public view returns(uint) { return result; }
function getData() public view returns(uint) { return c.info(); }
}
```



B) Inheritance:

Inheritance is a way to extend functionality of a contract. Solidity supports both single as well as multiple inheritance.

```
// Solidity program to
// demonstrate
// Single Inheritance
pragma solidity >=0.4.22 <0.6.0;

// Defining contract
contract parent{
// Declaring internal
// state variable
uint internal sum;

// Defining external function
// to set value of internal
// state variable sum
function setValue() external {
uint a = 20;
uint b = 20;
sum = a + b;
}

// Defining child contract
contract child is parent{

// Defining external function
```

```

// to return value of
// internal state variable sum
function getValue() external view returns(uint) {
return sum;
}
}

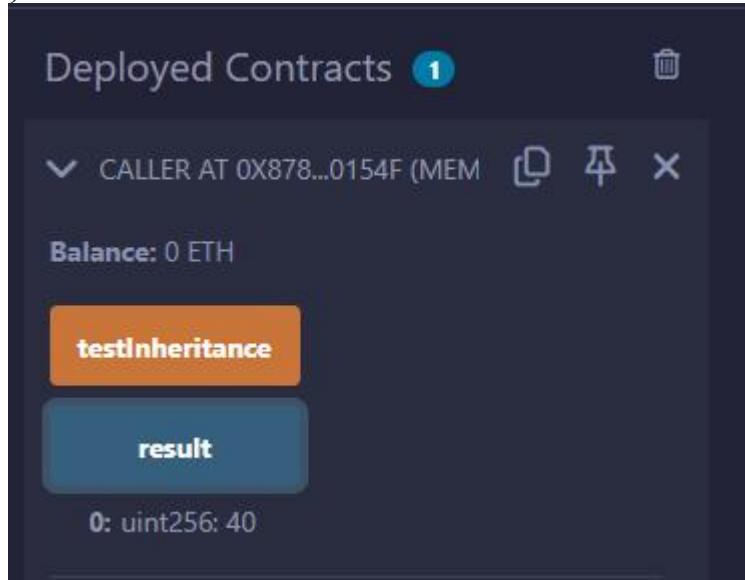
// Defining calling contract
contract caller {

// Creating child contract object
child cc = new child();

// Defining function to call
// setValue and getValue functions
function testInheritance() public {
cc.setValue();
}

function result() public view returns(uint){
return cc.getValue();
}
}

```



C) Constructors:

Constructor is a special function declared using constructor keyword. It is an optional function and is used to initialize state variables of a contract. Following are the key characteristics of a constructor.

A contract can have only one constructor.

A constructor code is executed once when a contract is created and it is used to initialize contract state.

A constructor can be either public or internal.

An internal constructor marks the contract as abstract.

In case, no constructor is defined, a default constructor is present in the contract.

```

pragma solidity ^0.5.0;
contract Base {
uint data;
constructor(uint _data) public {
data = _data;
}
function getResult() public view returns(uint) {

```

```
return data;  
}  
}  
contract Derived is Base (5) {  
constructor() public {}  
}
```

The screenshot shows the Truffle UI interface. At the top, there is a code editor with Solidity code. Below the code editor is a toolbar with a "Deploy" button and a "Publish to IPFS" checkbox. A "At Address" button is highlighted, showing the address 0x4B20993Bc481177ec7E8f57. The main area displays the deployed contracts. Under "Transactions recorded", there is a checkbox for "Run transactions using the latest compilation result" which is checked. Below this are "Save" and "Run" buttons. Under "Deployed Contracts", there is one entry labeled "BASE AT 0X13E...3F3E4 (MEMO)". The "Balance" for this contract is shown as 0 ETH. A "getresult" button is present, and the result of the call is displayed as "0: uint256: 6".

// Indirect Initialization of Base Constructor

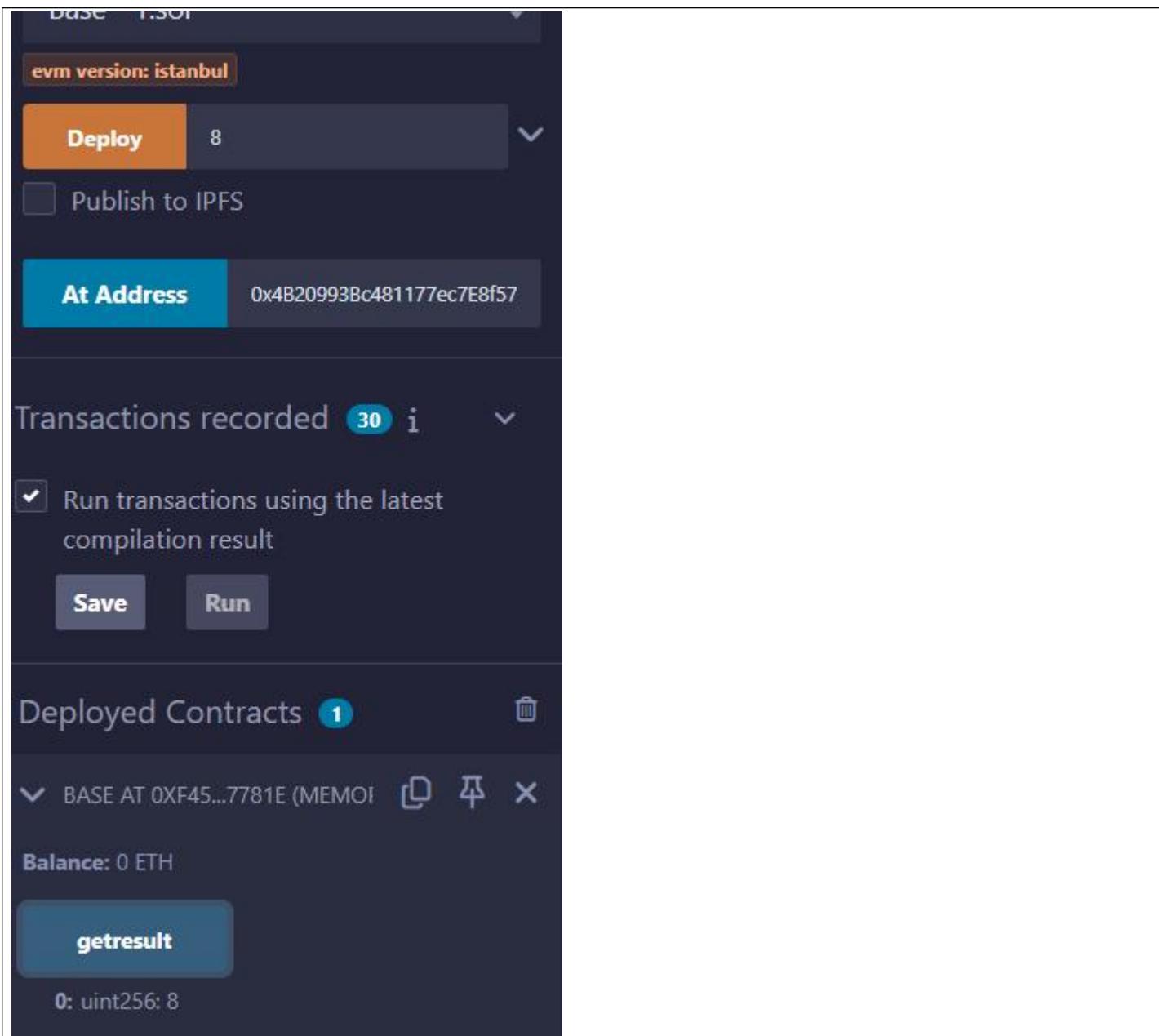
```
pragma solidity ^0.5.0;  
  
contract Base {  
uint data;  
constructor(uint _data) public {  
data = _data;  
}  
function getResult() public view returns(uint){  
return data;  
}  
}  
contract Derived is Base {  
constructor(uint _info) Base(_info * _info) public {}  
}
```

D)Abstract Contracts:

Abstract Contract is one which contains at least one function without any implementation. Such a contract is used as a base contract. Generally an abstract contract contains both implemented as well as abstract functions. Derived contract will implement the abstract function and use the existing functions as and when required.

```
pragma solidity ^0.5.0;
```

```
contract Calculator {  
function getResult() public view returns(uint);  
}  
contract Test is Calculator {  
function getResult() public view returns(uint) {  
uint a = 4;  
uint b = 2;  
uint result = a + b;  
return result;  
}  
}
```



E)Interfaces:

Interfaces are similar to abstract contracts and are created using interface keyword. Following are the key characteristics of an interface.

Interface can not have any function with implementation.

Functions of an interface can be only of type external.

Interface can not have constructor.

Interface can not have state variables.

```
pragma solidity ^0.5.0;
```

```
interface Calculator {
    function getResult() external view returns(uint);
}
contract Test is Calculator {
    constructor() public {}
    function getResult() external view returns(uint){
        uint a = 5;
        uint b = 2;
    }
}
```

```
uint result = a + b;  
return result;  
}  
}
```

VALUE
10 Wei

CONTRACT
Calculator - 1.sol

evm version: istanbul

Deploy

Publish to IPFS

At Address 0x4B20993Bc481177ec7E8f57

Transactions recorded 0 i

Run transactions using the latest compilation result

Save **Run**

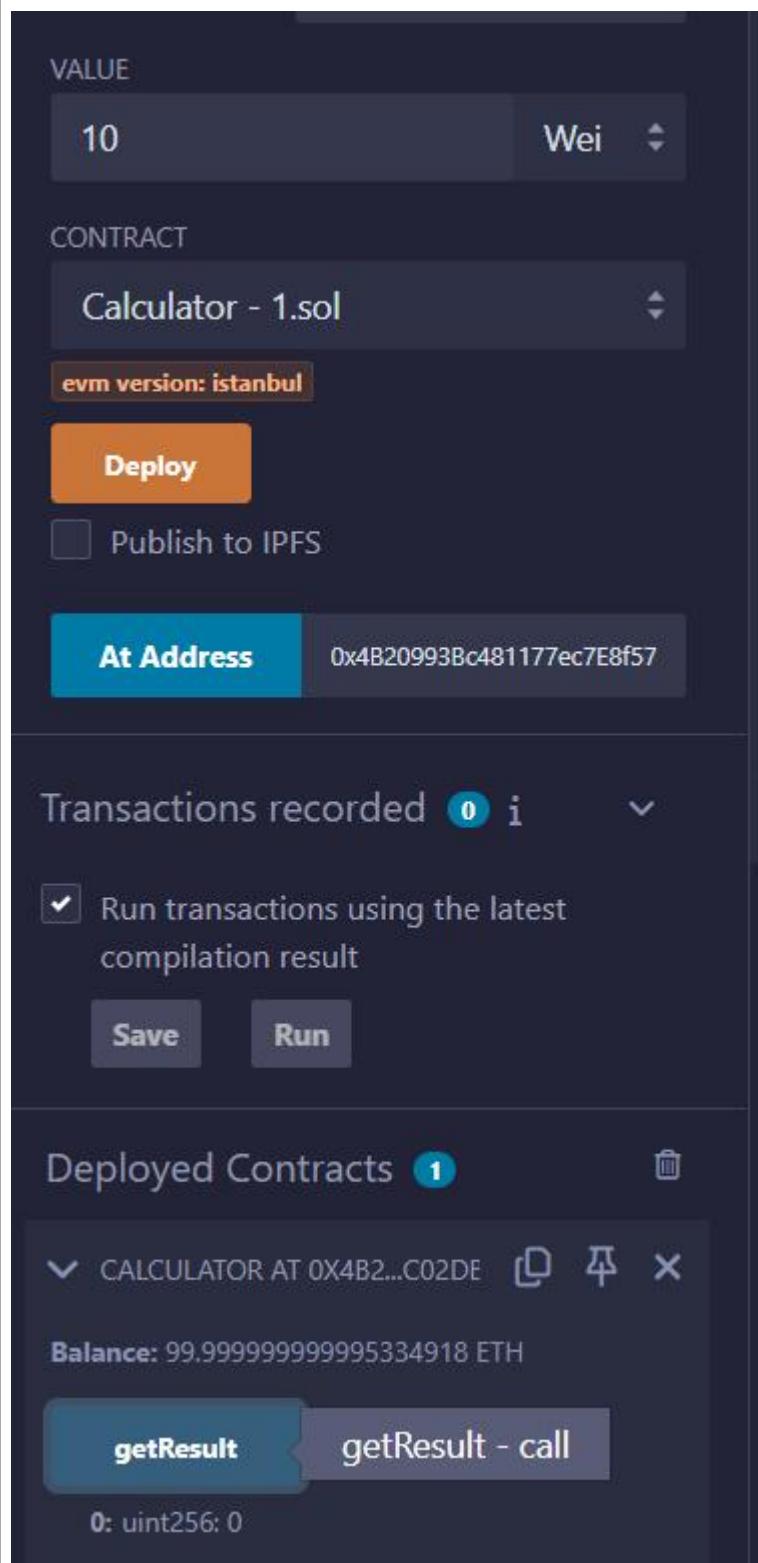
Deployed Contracts 1

CALCULATOR AT 0X4B2...C02DE   

Balance: 99.99999999995334918 ETH

getResult getResult - call

0: uint256: 0



PRACTICAL 6c

Aim:-Libraries,Assembly, Events, Error handling.

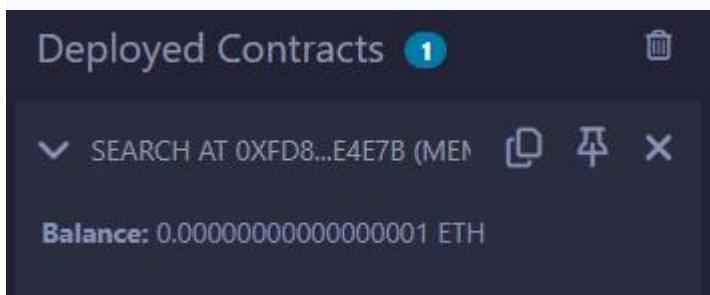
F) Libraries:

Libraries are similar to Contracts but are mainly intended for reuse. A Library contains functions which other contracts can call. Solidity have certain restrictions on use of a Library.

```
pragma solidity ^0.5.0;

library Search {
    function indexOf(uint[] storage self, uint value) public view returns (uint) {
        for (uint i = 0; i < self.length; i++)
            if (self[i] == value) return i;
        return uint(-1);
    }
}

contract Test {
    uint[] data;
    uint value;
    uint index;
    constructor() public {
        data.push(6);
        data.push(7);
        data.push(8);
        data.push(9);
        data.push(10);
    }
    function isValuePresent() external {
        value = 9;
        //search if value is present in the array using Library function
        index = Search.indexOf(data, value);
    }
    function getResult() public view returns(uint){
        return index;
    }
}
```



G) Assembly:

Solidity provides an option to use assembly language to write inline assembly within Solidity source code. We can also write a standalone assembly code which then be converted to bytecode. Standalone Assembly is an intermediate language for a Solidity compiler and it converts the Solidity code into a Standalone Assembly and then to byte code. We can used the same language used in Inline Assembly to write code in a Standalone assembly.

```
pragma solidity ^0.5.0;

library Sum {
    function sumUsingInlineAssembly(uint[] memory _data) public pure returns (uint o_sum) {
        for (uint i = 0; i < _data.length; ++i) {
```

```
assembly {
o_sum := add(o_sum, mload(add(add(_data, 0x20), mul(i, 0x20))))
}
}
}

contract Test {
uint[] data;
constructor() public {
data.push(1);
data.push(2);
data.push(3);
data.push(4);
data.push(5);
}
function sum() external view returns(uint){
return Sum.sumUsingInlineAssembly(data);
}
}
```

The screenshot shows the Truffle UI interface. At the top, there's a header bar with the title "Deployed Contracts" and a count of 1. Below this, a card displays a deployed contract named "SUM AT 0X7A5...58287 (MEMO)". The card includes a "Balance: 0 ETH" section and a "Low level interactions" section with a "Transact" button. In the bottom half of the screen, two transaction logs are listed:

- A log for the creation of the contract: [vm] from: 0x4B2...C02db to: Search.(constructor) value: 10 wei data: 0x610...10032 logs: 0 hash: 0x027...00254. It has a "Debug" button and a dropdown menu.
- A log for a call to the contract's constructor: [vm] from: 0x4B2...C02db to: Sum.(constructor) value: 0 wei data: 0x610...10032 logs: 0 hash: 0x97f...7b1c6. It also has a "Debug" button and a dropdown menu.

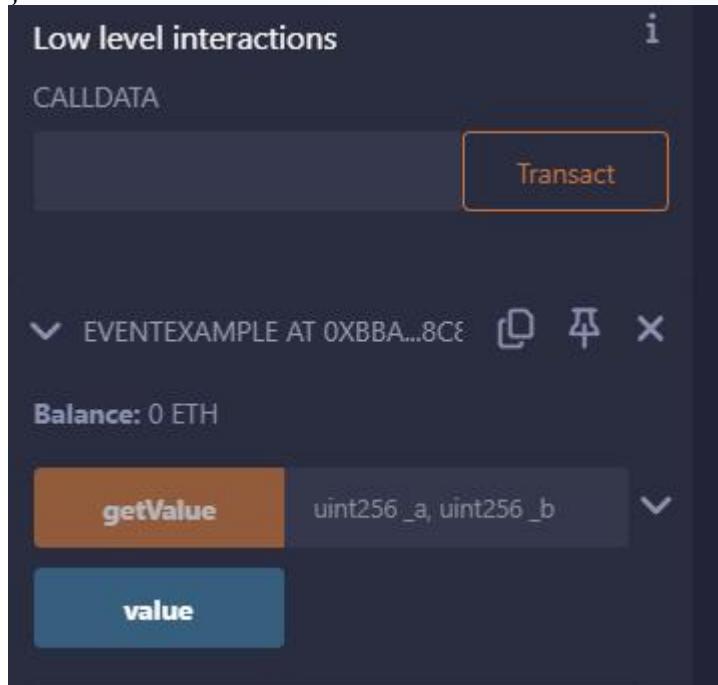
Below the logs, error messages indicate issues with argument encoding for the function calls:

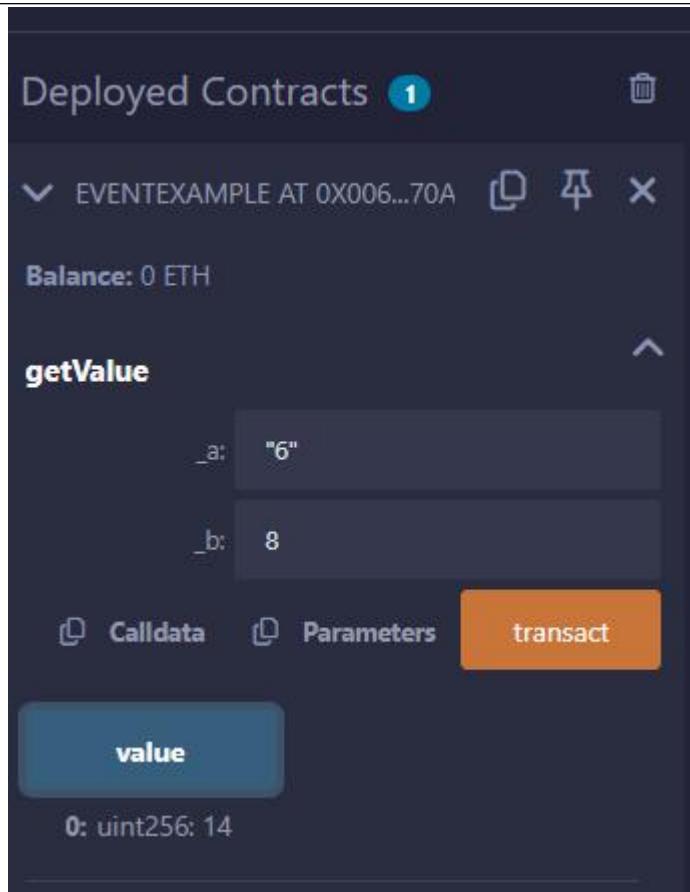
- "call to Sum.sumUsingInlineAssembly errored: Error encoding arguments: Error:"
- "call to Sum.sumUsingInlineAssembly errored: Error encoding arguments: Error:"

H)Events:

Event is an inheritable member of a contract. An event is emitted, it stores the arguments passed in transaction logs. These logs are stored on blockchain and are accessible using address of the contract till the contract is present on the blockchain. An event generated is not accessible from within contracts, not even the one which have created and emitted them.

```
// Solidity program to demonstrate  
// creating an event  
pragma solidity ^0.4.21;  
  
// 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;  
}  
}
```





I) Error Handling:

Solidity provides various functions for error handling. Generally when an error occurs, the state is reverted back to its original state. Other checks are to prevent unauthorized code access.

Solidity program to demonstrate require statement.

```
// Solidity program to
// demonstrate require
// statement

pragma solidity ^0.5.0;
// Creating a contract
contract requireStatement {
// Defining function to
// check input
function checkInput(uint8 _input) public view returns(string memory){
require(_input >= 0, "invalid uint");
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;
}
```

Deployed Contracts 1

REQUIRESTATEMENT AT 0XF16.

Balance: 0 ETH

checkInput

Odd uint256_input

REQUIRESTATEMENT AT 0XF16.

Balance: 0 ETH

checkInput

_input: "7"

Calldata Parameters

0: string: Input is UInt8

Odd

_input: 9

Calldata Parameters

0: bool: true

Solidity program to demonstrate assert statement.

```
// Solidity program to
// demonstrate assert
// statement

pragma solidity ^0.5.0;

// Creating a contract
contract assertStatement {
// Defining a state variable
bool result;
// Defining a function
// to check condition
function checkOverflow(uint8 _num1, uint8 _num2) public {
uint8 sum = _num1 + _num2;
assert(sum<=255);
```

```

result = true;
}
// Defining a function to
// print result of assert
// statement
function getResult() public view returns(string memory){
if(result == true){
return "No Overflow";
}
else{
return "Overflow exist";
}
}
}
}

```

Deployed Contracts 1

✓ ASSERTSTATEMENT AT 0X7FD...

Balance: 0 ETH

checkOverflow Input required 2 ▾

getResult

Deployed Contracts 1

✓ ASSERTSTATEMENT AT 0X7FD...

Balance: 0 ETH

checkOverflow

_num1: 6

_num2: 10

Calldata Parameters **transact**

getResult

0: string: No Overflow

Solidity program to demonstrate revert statement.

```

// Solidity program to
// demonstrate revert

```

```
pragma solidity ^0.5.0;
// Creating a contract
contract revertStatement {
// Defining a function
// to check condition
function checkOverflow(uint _num1, uint _num2) public view returns(
string memory, uint) {
uint sum = _num1 + _num2;
if(sum < 0 || sum > 255){
revert(" Overflow Exist");
}
else{
return ("No Overflow", sum);
}
}
}
}
https://www.tutorialspoint.com/solidity/index.htm
```

The image displays two vertically stacked screenshots of a blockchain development environment, likely Truffle or Remix, showing deployed contracts.

Top Screenshot: Shows a list of deployed contracts under "Deployed Contracts". One contract is expanded, showing its details. The contract name is "REVERTSTATEMENT AT 0X794...". It has a balance of 0 ETH. A transaction button labeled "checkOverflow" is visible, along with input fields for "_num1" and "_num2".

Bottom Screenshot: Shows the same deployed contract "REVERTSTATEMENT AT 0X794...". The "checkOverflow" transaction is selected, and its parameters are displayed. The inputs are set to _num1: 8 and _num2: 9. Below the inputs, there are buttons for "Calldata" and "Parameters", and a large blue "call" button.

Output: The output of the call shows two results: 0: string: No Overflow and 1: uint256: 17.

PRACTICAL 7

Aim:-

7 Deploying a contracts on an external blockchain by using Ganache and/or MyEtherwallet, Metamask

PRACTICAL 1

Aim:-

Code:-

Output:-

8. Deploy a local private blockchain over a network with Ethereum or Rust (VM)

PRACTICAL 8

Aim:-Create your own blockchain and demonstrate its use.

Install on Ubuntu via PPAs

The easiest way to install go-ethereum on Ubuntu-based distributions is with the built-in launchpad PPAs (Personal Package Archives). We provide a single PPA repository that contains both our stable and development releases for Ubuntu versions trusty, xenial, zesty and artful.

linux:

To enable our launchpad repository run:

Step 1: open new terminal

Step 2: on terminal type this command

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

#if above command gives error then run

```
#sudo apt-get install --reinstall ca-certificates
```

Step 3: install the stable version of go-ethereum:

```
sudo apt-get update
```

```
sudo apt-get install ethereum
```

Step 4: create new directory for storing blockchain data

```
mkdir myblockchain2
```

```
cd myblockchain2
```

```
geth account new --datadir data
```

```
mithilesh@mithilesh-virtual-machine:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:6 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy InRelease
Get:7 http://in.archive.ubuntu.com/ubuntu jammy amd64 DEP-11 Metadata [7,048 B]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [216 B]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.8 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Fetched 152 kB in 3s (55.8 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
mithilesh@mithilesh-virtual-machine:~$ sudo apt-get install ethereum
'
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  bootnode
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  ethereum
1 upgraded, 0 newly installed, 0 to remove and 536 not upgraded.
Need to get 1,454 B of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy/main amd64 ethereum amd64 1.15.0+build30732+jammy [1,454 B]
Fetched 1,454 B in 1s (1,669 B/s)
(Reading database ... 194923 files and directories currently installed.)
Preparing to unpack .../ethereum_1.15.0+build30732+jammy_amd64.deb ...
Unpacking ethereum (1.15.0+build30732+jammy) over (1.11.5+build28443+jammy) ...
Setting up ethereum (1.15.0+build30732+jammy) ...
>
>
> mkdir myblockchain3
> cd myblockchain3
> geth account new --datadir data
>
```

Step 5: Create genesis.json file

```
sudo nano genesis.json
```

```
{
  "config": {
    "chainId": 12345,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "berlinBlock": 0,
    "ethash": {}
  },
  "difficulty": "1",
  "gasLimit": "8000000",
  "alloc": {
    "7df9a875a174b3bc565e6424a0050ebc1b2d1d82": { "balance": "300000" },
    "Efaf4df069211972a7D2C3306d1F778a1603F10F": { "balance": "400000" }
  }
}
```

save the file -> ctrl +o to write -> {enter} save -> ctrl +x exit

The screenshot shows a Linux desktop environment with a dark theme. In the top-left corner, there's a dock with icons for Home, Dash, Activities, Terminal, and others. A terminal window titled 'blockchainVM' is open, showing the command 'sudo nano genesis.json' and its contents. Below the terminal is a file editor window titled 'genesis.json' with the same JSON code. The desktop background is red, and there are several files on the desktop: 'myblockchain', 'private network.txt', 'hyperledger iroha.txt', 'bitcoinapi.txt', and 'dapp.txt'. The bottom status bar shows the date and time as 'Feb 13 15:45'.

```

{
  "config": {
    "chainId": 12345,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "berlinBlock": 0,
    "ethash": {}
  },
  "difficulty": "1",
  "gasLimit": "8000000",
  "alloc": {
    "7df9a875a174b3bc565e6424a0050ebc1b2d1d82": { "balance": "300000" },
    "Efaf4df069211972a7D2C3306d1F778a1603F10F": { "balance": "400000" }
  }
}

```

Step 6: initialize the block

geth init --datadir data genesis.json

Step 7: create network

geth --datadir data --networkid 12345

[do not close this terminal]

||||||||||||||||||||||||||||||||||||||||||||

```

mithilesh@mithilesh-virtual-machine: ~
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy InRelease [127 kB]
Hit:6 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy DEP-11 Metadata [7,048 B]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [216 B]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [216 B]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.8 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Fetched 152 kB in 3s (55.8 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  bootnode
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  ethereum
1 upgraded, 0 newly installed, 0 to remove and 536 not upgraded.
Need to get 1,454 B of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy/main amd64 ethereum amd64 1.15.0+build30732+jammy [1,454 B]
Fetched 1,454 B in 1s (1,669 B/s)
(Reading database ... 194923 files and directories currently installed.)
Preparing to unpack .../ethereum_1.15.0+build30732+jammy_amd64.deb ...
Unpacking ethereum (1.15.0+build30732+jammy) over (1.11.5+build28443+jammy) ...
Setting up ethereum (1.15.0+build30732+jammy) ...
>
> mkdir myblockchain
> cd myblockchain
> geth account new --datadir data
> sudo nano genesis.json
> geth init --datadir data genesis.json
> geth --datadir data --networkid 12345

```

Step 8: open new tab/terminal 2:

```

sudo geth attach data/geth.ipc
eth.getBalance(eth.accounts[0])
miner.setEtherbase(eth.accounts[0])
miner.start()
admin.addPeer(admin.nodeInfo.enode)
eth.getBalance(eth.accounts[0])

```

Step 10: Wait for 10-20 minutes and check balance

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

if ether balance is 0 wait for 10-20minutes for mining process to get complete and run eth.getBalance(eth.accounts[0]) again.

After balance is updated you can check current block height
eth.blockNumber

PRACTICAL 9

Aim:-

Code:-

Output:-

9Implement the mining module of Bitcoin client . The mining module, or miner, should produce blocks that solve proof-of-work puzzle

9. Compile and test smart contracts on a testing framework using the Ethereum Virtual Machine (EVM).

PRACTICAL 10

Aim:-
Code:-
Output:-

11 Demonstrate the running of the blockchain node

PRACTICAL 11

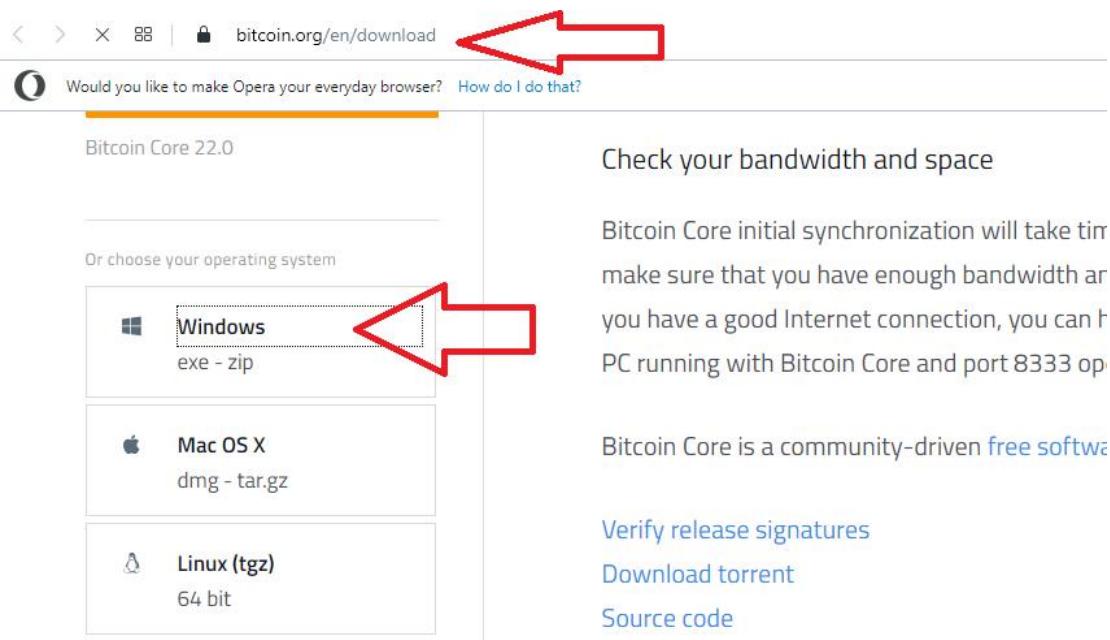
Aim:-
Code:-
Output:-

PRACTICAL 12

Aim:-Demonstrate the use of Bitcoin Core API.

Step 1: Visit: <https://bitcoin.org/en/download>

Step 2: Download windows setup [use and try with Linux version as well]



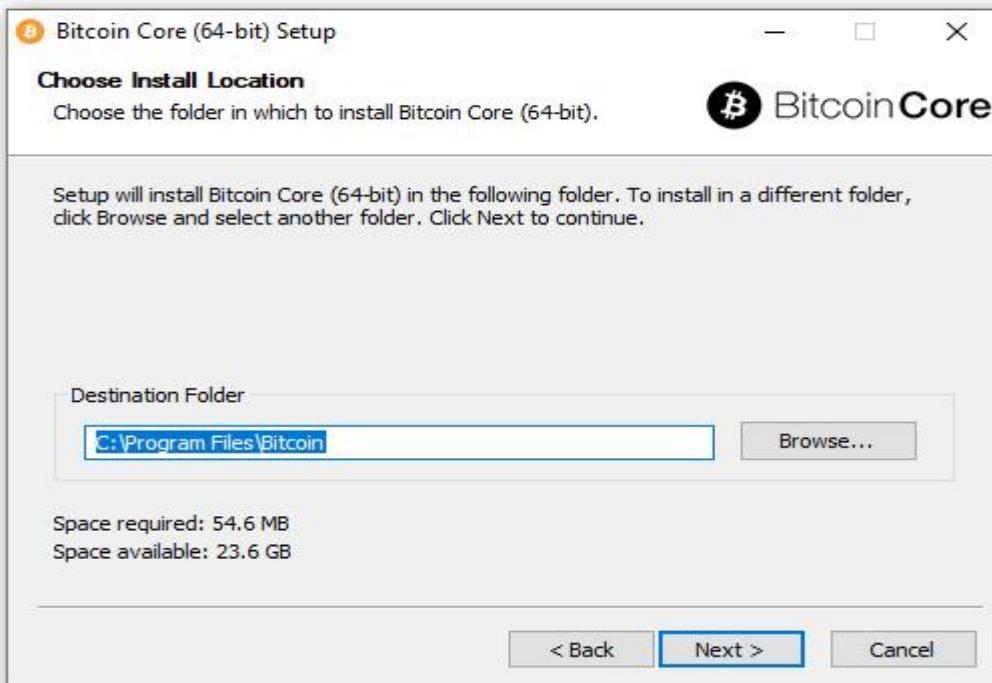
The screenshot shows the Bitcoin Core download page. At the top, there's a browser-like header with back, forward, and search buttons, followed by the URL 'bitcoin.org/en/download'. Below this, a message asks if you want to make Opera your everyday browser with a link to 'How do I do that?'. The main content area has a yellow header 'Bitcoin Core 22.0'. Underneath, it says 'Or choose your operating system' and lists three options: 'Windows' (selected), 'Mac OS X', and 'Linux (tgz)'. The 'Windows' option is highlighted with a red box and a red arrow pointing to it from the left. To the right of the operating system choices, there's a section titled 'Check your bandwidth and space' with instructions about initial synchronization. Further down, there's a note about Bitcoin Core being free software, and links for 'Verify release signatures', 'Download torrent', and 'Source code'.

Step 3: Run the setup file-> click

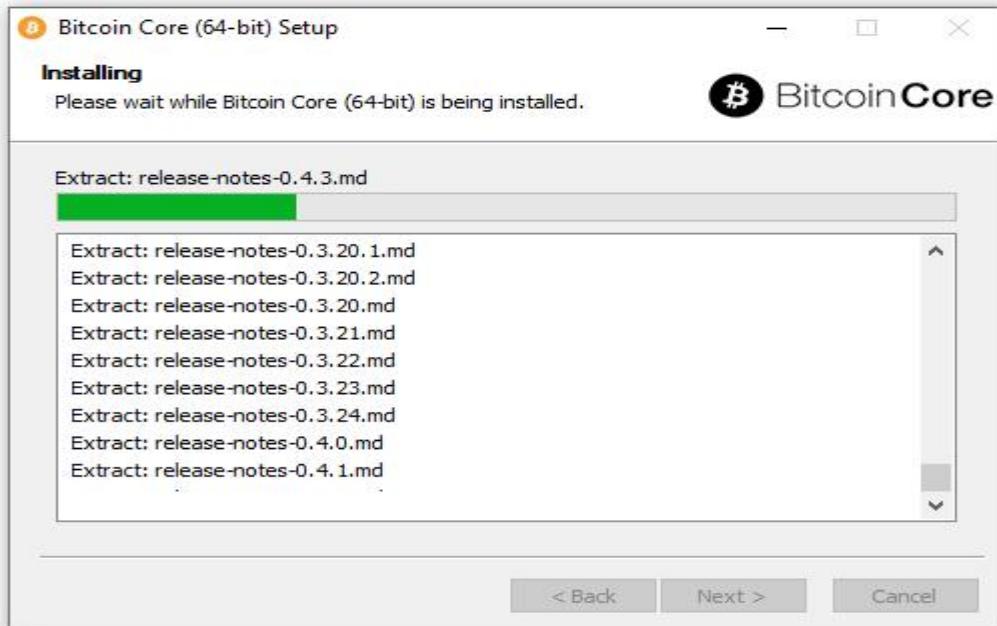
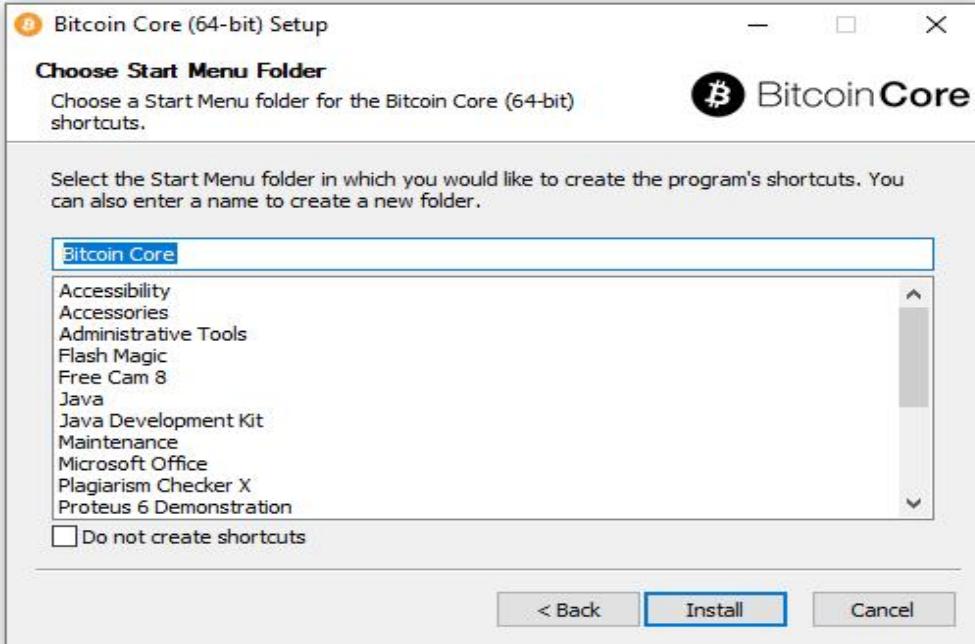


next

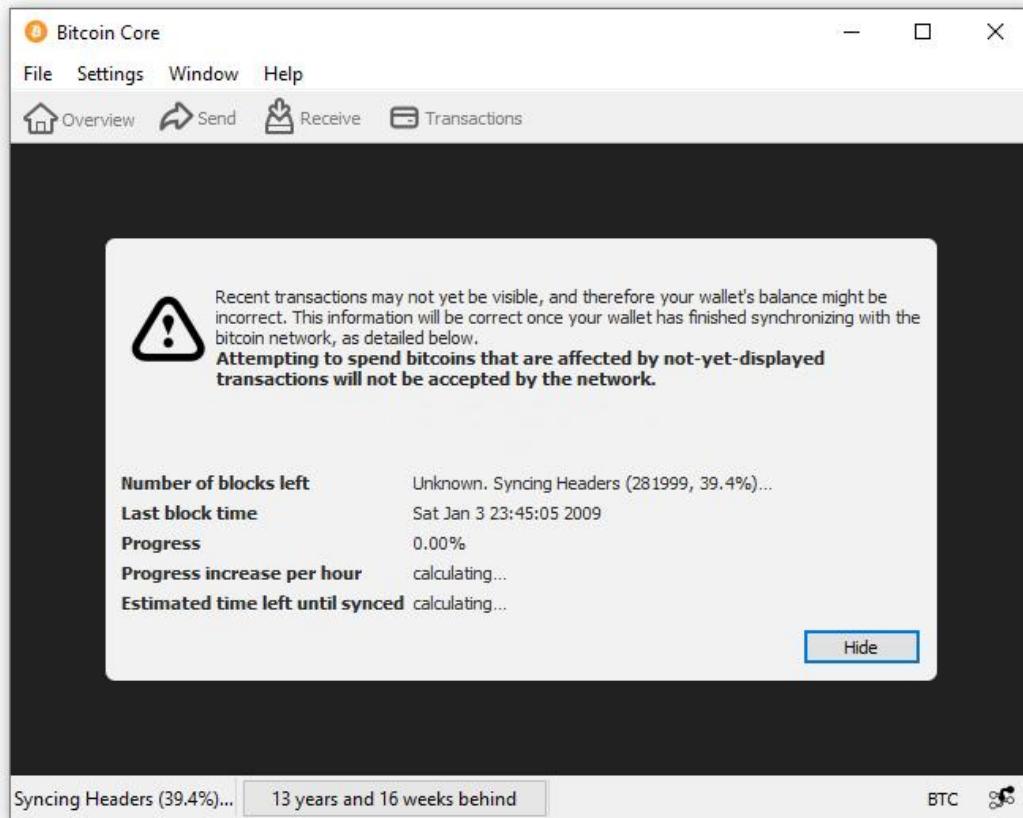
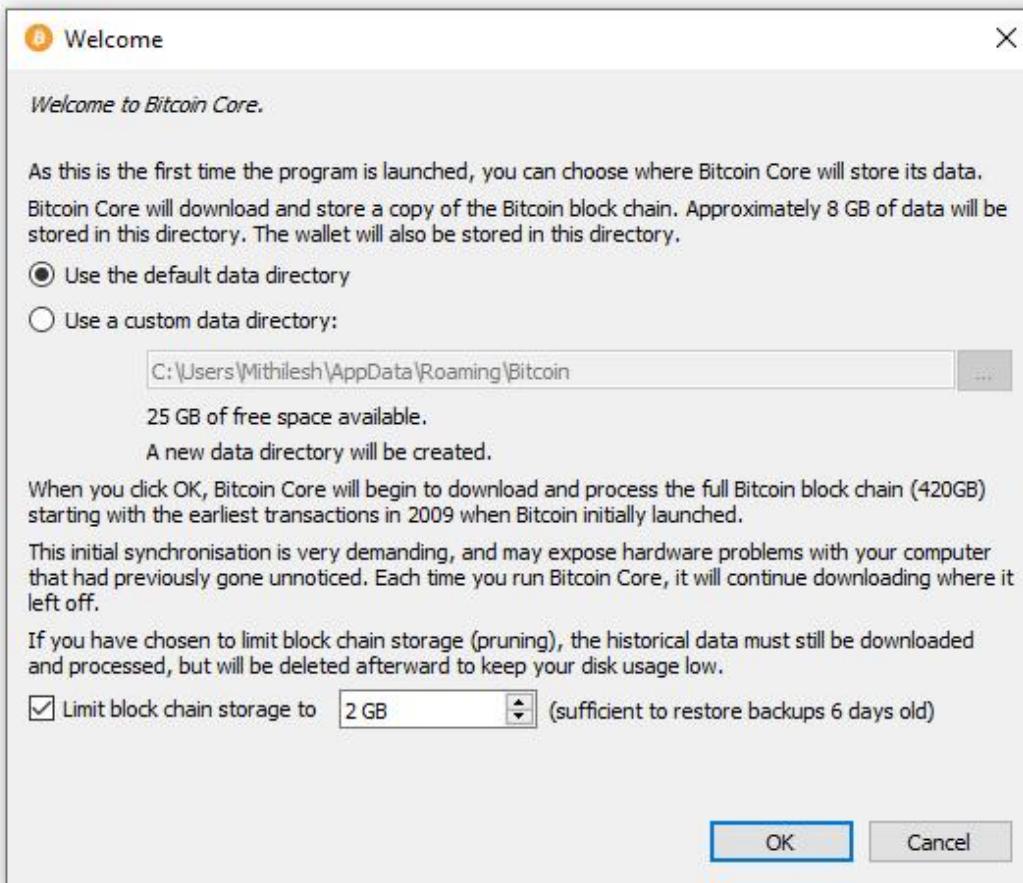
Step 4: Click Next



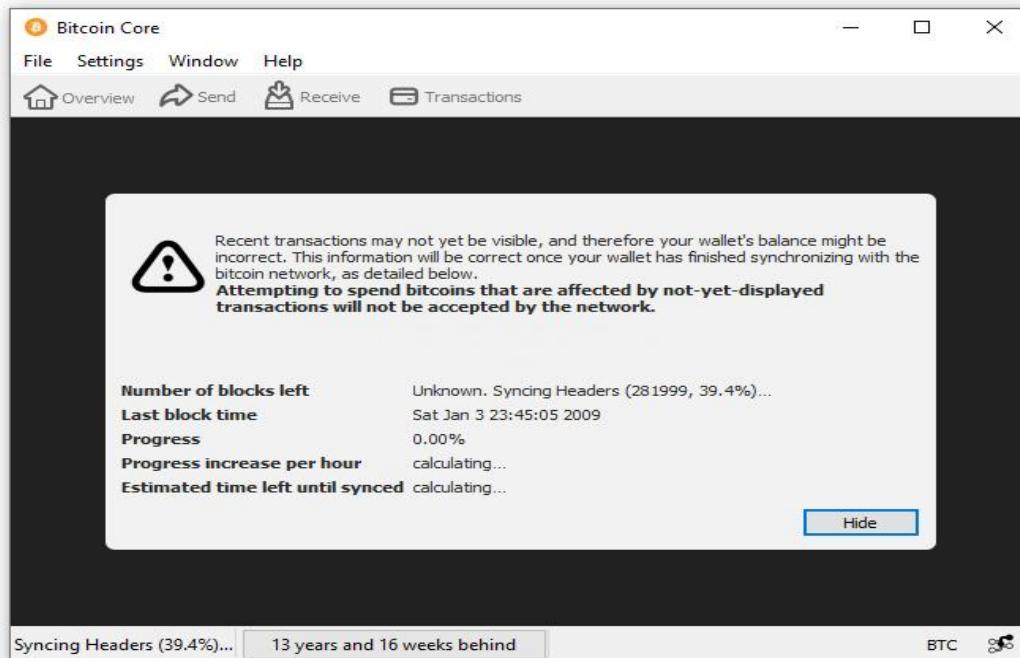
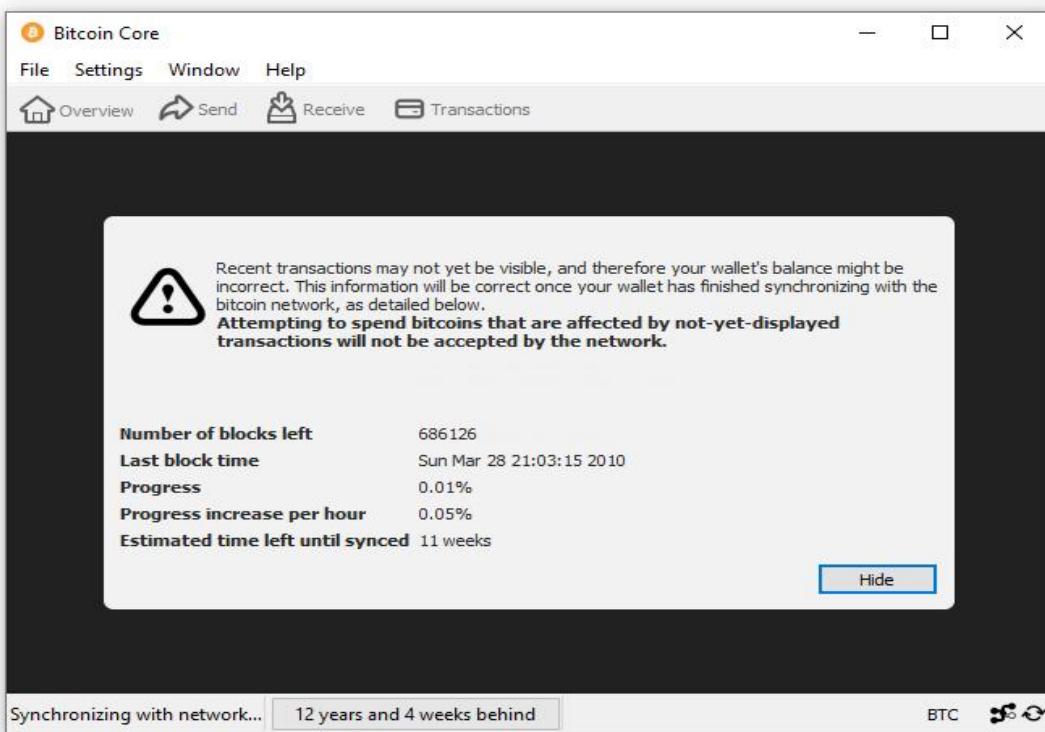
Step 5: Finally click on Install



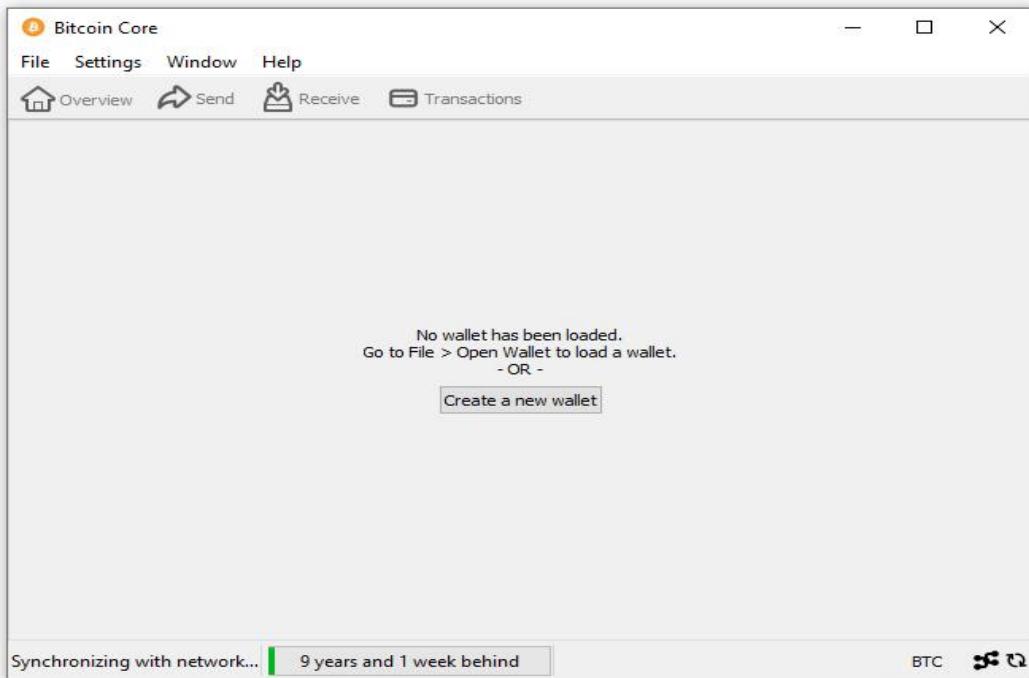
Launch Bitcoin Core-> Click OK.



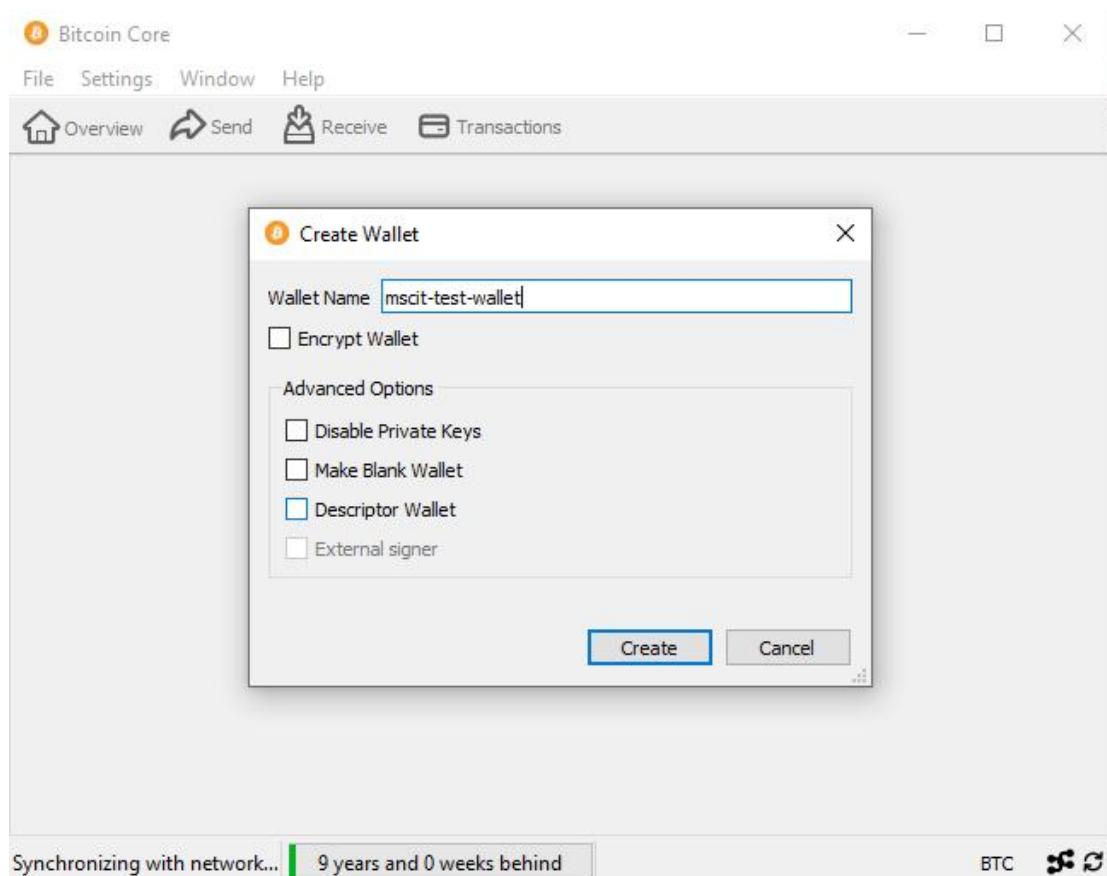
Click on Hide button [Synchronization take place in background]



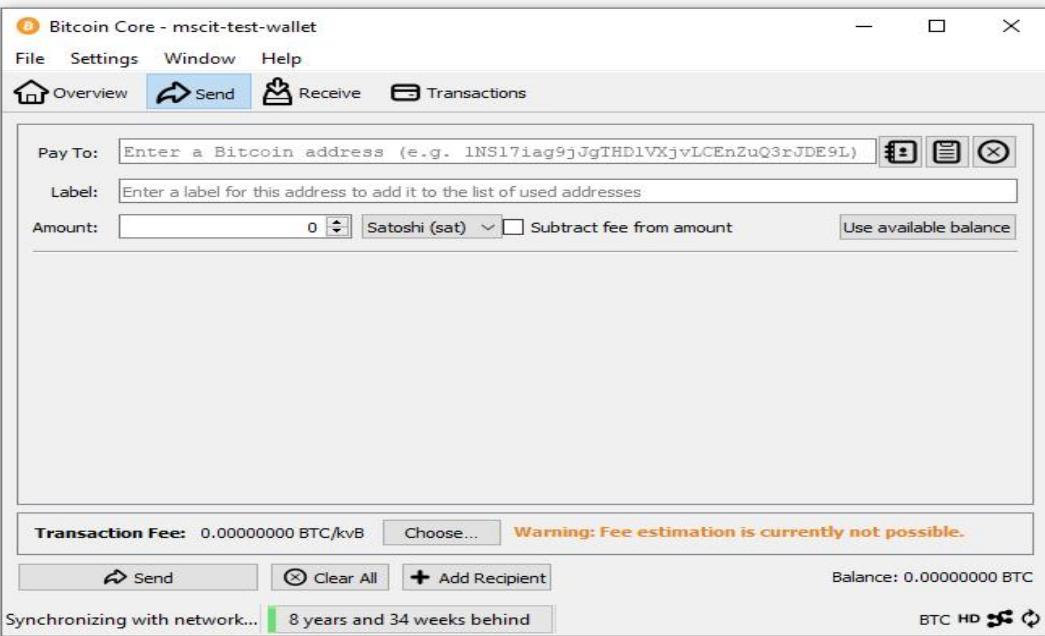
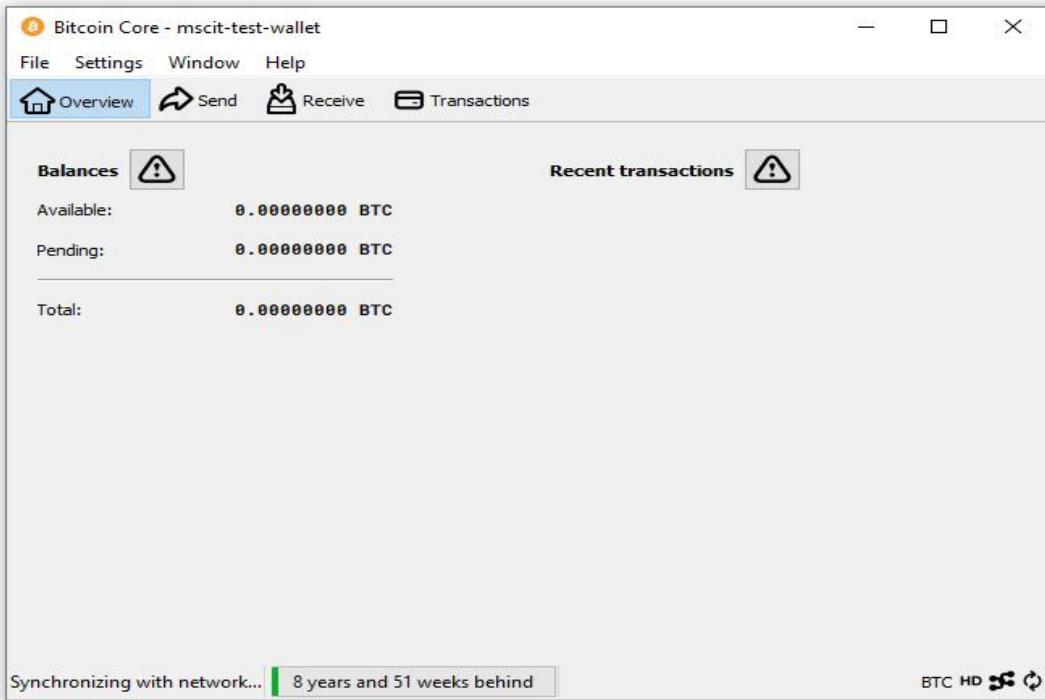
You can create a wallet -> Create a new wallet

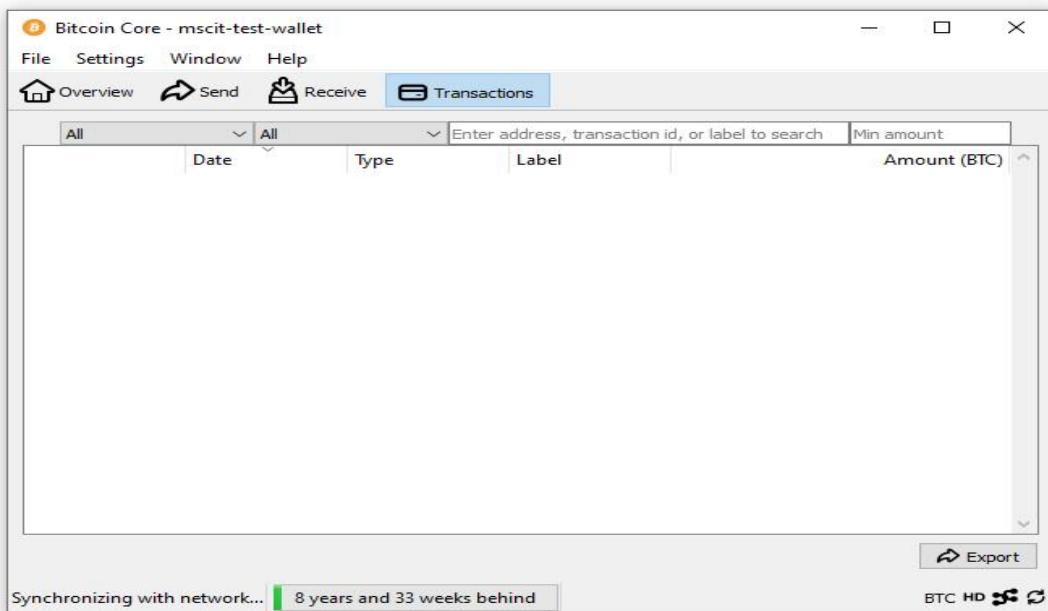
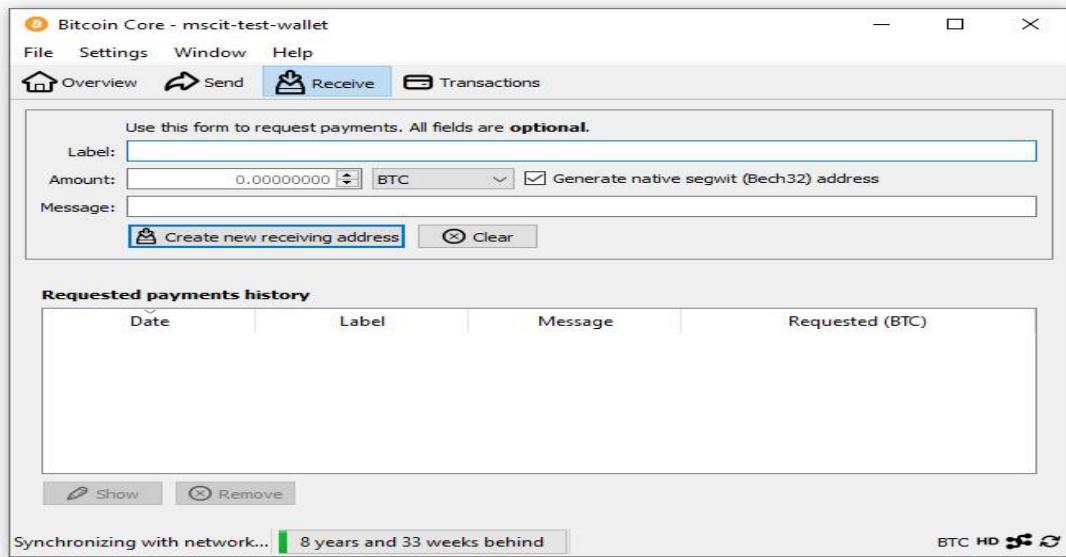


Enter Wallet name



Finally Account is setup





Open Python IDLE and create new Script.

```
#####
from bitcoinlib.wallets import Wallet
w = Wallet.create('Wallet1')
key1 = w.get_key()
print('Wallet Address:',key1.address)
w.scan()
print(w.info())
```

```

Wallet Address: bc1qppnqpg9quay7qf5hz2ekx2cu0g8tmky666u5n
== WALLET ==
ID 1
Name Wallet1
Owner
Scheme bip32
Multisig False
Witness type segwit
Main network bitcoin
Latest update 2025-02-17 04:17:53.572643+00:00

= Wallet Master Key =
ID 1
Private True
Depth 0

- NETWORK: bitcoin -
- Keys
  6 m/84'/0'/0'/0/0      bc1qppnqpg9quay7qf5hz2ekx2cu0g8tmky666u5n address index 0 0.0000000 ฿
  7 m/84'/0'/0'/0/1      bc1qy762wx0jqp9y6pseskwhg6yn2h0z2ry7gavymra address index 1 0.0000000 ฿
  9 m/84'/0'/0'/0/2      bc1q0lvwm045pl0q9hdvucwk8h5dmvu88grdeuyj address index 2 0.0000000 ฿
  10 m/84'/0'/0'/0/3     bc1ql33pg2mjzempsyzxme740xzssln7fc7hhry4fl address index 3 0.0000000 ฿
  11 m/84'/0'/0'/0/4     bc1q4p3fhm8r7sjwzhha9lrxp5wg0thapn2k3xj5fe address index 4 0.0000000 ฿
  13 m/84'/0'/0'/1/0    bc1qkcus1wjmd2uhgs9auuuua0gydt4sc6jrljqesx address index 0 0.0000000 ฿
  15 m/84'/0'/0'/1/1    bc1qs29svsg7he7l2nqpk6y4k0x5hk5zev9r0nv5 address index 1 0.0000000 ฿
  16 m/84'/0'/0'/1/2    bc1qehe7dpnxnd55rc35tgg7qr2c4krah70pv2wv address index 2 0.0000000 ฿
  17 m/84'/0'/0'/1/3    bc1q3cwgkrmrcclprvd9caqpfjhnyzcnnhw0pag3 address index 3 0.0000000 ฿
  18 m/84'/0'/0'/1/4    bc1q3wd8tnqsy7a5u7rugz07de54cwdx8vdswn2fxr address index 4 0.0000000 ฿

- - Transactions Account 0 (0)

= Balance Totals (includes unconfirmed) =

None
#####

```

Open CMD and install **bitcoinlib** package

pip install bitcoinlib

PRACTICAL 13

Aim:-Create your own blockchain and demonstrate its use.

Code:-

```
# following imports are required by PKI
```

```
import hashlib
import random
import binascii
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 = ""
```

```
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
    for i in range(1000):
        digest = sha256(str(hash(message)) + str(i))
        if digest.startswith(prefix):
            return i #i= nonce value

A = Client()
B =Client()
C =Client()
t0 = Transaction (
    "Genesis",
    A.identity,
    500.0
)
t1 = Transaction (
    A,
    B.identity,
    40.0
)
t2 = Transaction (
    A,
    C.identity,
    70.0
)
t3 = Transaction (
    B,
    C.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)
```

```
#####
#####
```

```
save the file -> ctrl +O to write -> {enter} save -> ctrl +x exit
```

Run this file

Output:

```
Number of blocks in the chain: 3
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100b1558beccb109cbf1c223ebf6e791ea734e20c03dc8bce926f3b28c9e2a2383cf4ae87b6431211299b11
7de6143373a6682a64e0c7de6955fb9dc8806103d4c6a738d92d7511112e944c9d6eb51730b76e2b0b6a48069b6bd90c52549832429cbf8ba7ade362d4f3b04a5d568f54d30d6e3cb57ceaf1
8e7e7b2f2df2d8d2530203010001
-----
value: 500.0
-----
time: 2025-02-17 10:34:26.627963
-----
-----
=====
block # 1
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100b1558beccb109cbf1c223ebf6e791ea734e20c03dc8bce926f3b28c9e2a2383cf4ae87b6431211299b117d6
e143373a6682a64e0c7de6955fb9dc8806103d4c6a738d92d7511112e944c9d6eb51730b76e2b0b6a48069b6bd90c52549832429cbf8ba7ade362d4f3b04a5d568f54d30d6e3cb57ceaf18e7
e7b2f2d2df2d8d2530203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100bbcfc0e76057d12120ce04cd708cc9dca881bca54eaf520584faadeaa97a6b90de616b04036b465e8f1b
b79c3236f59d83c818830280b108c1a114026eb93cff404b9b78d757147b83cc259e099ae9166a0afde96a06f6fdcd1c8d99aab21ded8bf89ea7eab67efdbd52370daf555eee8ceb7e56a2
ba7c37f441c6ac00aa70203010001
-----
value: 40.0
-----
time: 2025-02-17 10:34:26.627963
-----
-----
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100b1558beccb109cbf1c223ebf6e791ea734e20c03dc8bce926f3b28c9e2a2383cf4ae87b6431211299b117d6
e143373a6682a64e0c7de6955fb9dc8806103d4c6a738d92d7511112e944c9d6eb51730b76e2b0b6a48069b6bd90c52549832429cbf8ba7ade362d4f3b04a5d568f54d30d6e3cb57ceaf18e7
e7b2f2d2df2d8d2530203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100bcfacd1614746e02a0ab42490f4932f55c8a339b2e2c4661c74950d9a1a22c3217e56525342d57bc51f2
022fd48e63f476916d50203010001
-----
value: 70.0
-----
time: 2025-02-17 10:34:26.627963
-----
-----
=====
block # 2
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100bbcfc0e76057d12120ce04cd708cc9dca881bca54eaf520584faadeaa97a6b90de616b04036b465e8f1bbb2
b79c3236f59d83c818830280b108c1a114026eb93cff404b9b78d757147b83cc259e099ae9166a0afde96a06f6fdcd1c8d99aab21ded8bf89ea7eab67efdbd52370daf555eee8ceb7e56a2b7
c37f441c6ac00aa70203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100bcfacd1614746e02a0ab42490f4932f55c8a339b2e2c4661c74950d9a1a22c3217e56525342d57bc51f2
c5a97fe5bd0601c8e6ecc30b5752c3c6c927add09703245b10ee469ae427d4929d947e51d4e1b8921cf2fb83f6d20aef8be123f9a30ac2d8eb602a8fdcf2dfb8150db16a31196af78de1b09028
022fd48e63f476916d50203010001
-----
value: 700.0
-----
time: 2025-02-17 10:34:26.627963
-----
=====
```

EXTRA

PRACTICAL 1

Aim:-

Code:-
Output:-

<https://tinyurl.com/vaze123>
format of practical

Practical No.

Aim: Question

Program:

Output:

Practical 1

aim: types of variable

```
pragma solidity ^0.5.0;
contract Pract1 {
    int x=15; //state var
    int public y=10;//global
    function getValue(int z) public{
        y=y+z;
    }

    function show() public view returns (int)
    {
        return x;
    }
}
```

Practical 2

Aim: relational operators

```
pragma solidity ^0.5.0;
contract Pract2 {
    bool public a=true;
    bool public b=false;
    bool public r1or=a||b;
    bool public r2and=a&&b;
    bool public r3not=!b;
}
```

Practical 3

aim:for loop

```
pragma solidity ^0.5.0;
contract Pract3 {
    function test(int s, int e) public view returns(int)
    {
        int i;
        int sum=0;
        for(i=s;i<=e;i++)
        {
            sum+=i; //sum=sum+i;
        }
        return sum;
    }
}
```

Practical 4

Aim: while loop

```
pragma solidity ^0.5.0;
contract Pract3{
function test(int s, int e) public view returns(int)
{
int i;
int sum=0;
i=s;
while(i<=e)
{
sum+=i; //sum=sum+i;
i++;
}
return sum;
}
```

Practical 5

Aim: do while loop

```
pragma solidity ^0.5.0;
contract Pract3{
function test(int s, int e) public view returns(int)
{
int i;
int sum=0;
i=s;
do
{
sum+=i; //sum=sum+i;
i++;
}while(i<=e);
return sum;
}
}
```

Practical 6

Aim: if else

```
pragma solidity ^0.5.0;
contract Pract4{
function test(int x) public view returns(string memory)
{
if(x%2==0)
return "Number is even";
else
return "Number is odd";
}
}
```

Practical 7

Aim: string

```
pragma solidity ^0.5.0;
contract Pract4{
function test(int x) public view returns(string memory)
{
if(x%2==0)
return "Number is even";
else
}
```

```

return "Number is odd";
}
}
}

Practical 8
Aim:array
contract Types {
uint[5] data;
constructor() public
{
    data = [uint(10), 20, 30, 40, 50];
}
function array_example() public view returns (uint,uint)
{
    return (data[0],data[4]);
}
function array_example2() public view returns (uint [5] memory)
{
    return data;
}
}

```

Practical 9

```

Aim: enum
pragma solidity ^0.5.0;
contract Types {
enum week_days
{
Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,
Sunday
}
week_days choice;
function set_value() public {
choice = week_days.Thursday;
}
function get_choice() public view returns (week_days) {
return choice;
}
}

```

Practical 10

//Aim: arithmetic operations

```
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;
}

```

Practical 11

Aim: structure

```

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;
    }
    function getBookDetail() public view returns (string memory, string memory,uint) {
        return (book.title, book.author, book.book_id);
    }
}

```

Practical 12

Aim: type of function(view, pure)

```
pragma solidity ^0.5.0;
contract Test {
int public x=10; //global
int y=90;//state
function f1() public returns(int){
    //read and update is allowed
    x=100;
return x;
}
function f2() public view returns(int){
    // x=100; //erro beacuse x is global/state
    //we can access but we cannot update state or global variable int view function
return x;
}
function f3() public pure returns(int){
    //we cannot access or update state or global variable in pure function
    int z=80;
return z;
}
}
```

Practical 13

Aim:function overloading

```
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;
}
}
```

Practical 14

Aim: mathematical function

```
pragma solidity ^0.5.0;

contract Test {
function callAddMod() public pure returns(uint){
return addmod(4, 5, 3);
}
function callMulMod() public pure returns(uint){
return mulmod(4, 5, 3);
}
}
```

Practical 15

Aim: cryptographic function

```
pragma solidity ^0.5.0;
contract Test {
function callKeccak256() public pure returns(bytes32 result){
return keccak256("ABC");
}
}
```

