**Name: Drishti Bhatia** **Roll Number: 07**
**SYMCA** **Block Chain Lab**

# Practical No: 08
# Aim: Smart Contract using Truffle Framework.

**Theory:**

## ❖ Truffle framework:

- Truffle is a world-class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life asa developer easier.
- Truffle is widely considered the most popular tool for blockchain application development with over 1.5 million lifetime downloads. Truffle supports developers across the full lifecycle of their projects, whether they are looking to build on Ethereum,Hyperledger, Quorum, or one of an ever-growing list of other supported platforms.
- Paired with Ganache, a personal blockchain, and Drizzle, a front-end dApp development kit, the full Truffle suite of tools promises to be an end-to-end dAppdevelopment platform.
  1. Built-in smart contract compilation, linking, deployment and binary management.
  2. Automated contract testing for rapid development.
  3. Scriptable, extensible deployment & migrations framework.
  4. Network management for deploying to any number of public & private networks.
  5. Package management with EthPM & NPM, using the ERC190 standard.
  6. Interactive console for direct contract communication.
  7. Configurable build pipeline with support for tight integration.
  8. External script runner that executes scripts within a Truffle environment.

**You can install Truffle with NPM in your command line like this:**

```
$ npm install -g truffle
```

## ❖ Smart Contract:

A smart contract is a stand-alone script usually written in Solidity and compiled intobinary or JSON and deployed to a specific address on the blockchain. In the same way
that we can call a specific URL endpoint of a RESTful API to execute some logic through an HttpRequest, we can similarly execute the deployed smart contract at a specific address by submitting the correct data along with the necessary Ethereum tocall the deployed and compiled Solidity function.

**Name: Drishti Bhatia** **Roll Number: 07**
**SYMCA** **Block Chain Lab**

1. **Install Node JS and Truffle Suite to develop and migrate the smart contracts into the Private Blockchain network. Make use of Truffle tools like compile, migrate and test for compilation, migration and testing the smart contracts through Blockchain**

**> npm install -g truffle**

```
C:\Users\EXAM>npm install -g truffle
npm WARN deprecated @types/keyv@4.2.0: This is a stub types definition. keyv provides its own type definitions, so you d
o not need this installed.
npm WARN deprecated mkdirp-promise@5.0.1: This package is broken and no longer maintained. 'mkdirp' itself supports prom
ises now, please switch to that.
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated multicodec@1.0.4: This module has been superseded by the multiformats module
npm WARN deprecated uuid@2.0.1: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain
 circumstances, which is known to be problematic.  See https://v8.dev/blog/math-random for details.
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain
 circumstances, which is known to be problematic.  See https://v8.dev/blog/math-random for details.
[               ] \ reify:uuid: WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions
```

**> mkdir blockchain-toolkit**

```
C:\Users\EXAM>mkdir blockchain-toolkit
```

**> cd blockchain-toolkit**

```
C:\Users\EXAM>cd blockchain-toolkit
```

**> truffle init**

```
C:\Users\EXAM\blockchain-toolkit>truffle init

Starting init...
================

> Copying project files to C:\Users\EXAM\blockchain-toolkit

Init successful, sweet!

Try our scaffold commands to get started:
  $ truffle create contract YourContractName # scaffold a contract
  $ truffle create test YourTestName         # scaffold a test

http://trufflesuite.com/docs
```

**> touch package.json**

```
C:\Users\EXAM\blockchain-toolkit>touch package.json
Touching package.json
```

**copy-and-pasting the below code into package.json file**

```
{
"name": "blockchain-toolkit",
"version": "1.0.0",
"description": "The Complete Blockchain Developer Toolkit for 2019 & Beyond",
"main": "truffle-config.js",
"directories": {
"test": "test"
},
"scripts": {
"dev": "lite-server",
"test": "echo \"Error: no test specified\" && sexit 1"
},
"author": "gregory@dappuniversity.com",
"license": "ISC",
"devDependencies": {
"bootstrap": "4.1.3",

"chai": "^4.1.2",
"chai-as-promised": "^7.1.1",
"chai-bignumber": "^2.0.2",
"dotenv": "^4.0.0",
"ganache-cli": "^6.1.8",
"lite-server": "^2.3.0",
"nodemon": "^1.17.3",
"solidity-coverage": "^0.4.15",
"truffle": "5.0.0-beta.0",
"truffle-contract": "3.0.6",
"truffle-hdwallet-provider": "^1.0.0-web3one.0"
}
}
```

**Save package.json file**
**Start developing a smart contract using solidity**

**> touch ./contracts/MyContract.sol**

```
C:\Users\EXAM\blockchain-toolkit>touch ./contracts/MyContract.sol
Touching ./contracts/MyContract.sol
```

**Name: Drishti Bhatia** **Roll Number: 07**
**SYMCA** **Block Chain Lab**

**Copy the below contract code and save in Mycontract.sol**

```
pragma solidity >=0.4.2 <=0.8.17;
contract MyContract {
string value;
constructor() public {
value = "myValue";
}
function get() public view returns(string memory) {
return value;
}
function set(string memory _value) public {
value = _value;
}
 }
```

**> truffle compile**

```
C:\Users\drish\blockchain-toolkit>truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\MyContract.sol
> Compiling .\contracts\MyContract.sol
> Artifacts written to C:\Users\drish\blockchain-toolkit\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang

C:\Users\drish\blockchain-toolkit>truffle migrate

Compiling your contracts...
===========================
> Compiling .\contracts\MyContract.sol
> Compiling .\contracts\MyContract.sol
> Artifacts written to C:\Users\drish\blockchain-toolkit\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
```

```
Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


2_deploy_contracts.js
=====================

   Replacing 'MyContract'
   ----------------------
   > transaction hash:    0xf423888bf4ab973d67d14eb574f8759286b9901087c5839f71b4866f6e005e85
   > Blocks: 0            Seconds: 0
   > contract address:    0xd46c21c2F6346Bf1D1E1eF9172211E6AB2dc4F03
   > block number:        20
   > block timestamp:     1667062354
   > account:             0xA707788b2E9D476307679406c2a5710684a9fd20
   > balance:             99.9275677
   > gas used:            392224 (0x5fc20)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00784448 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00784448 ETH
```

**Update the project configuration file (Find the file truffle-config.js and paste the following code)**

```
module.exports = {
networks: {
development: {
host: "127.0.0.1",
port: 7545,
network_id: "*" // Match any network id
}
},
solc: {
optimizer: {

enabled: true,
runs: 200
}
}
}
```

**Name: Drishti Bhatia** **Roll Number: 07**
**SYMCA** **Block Chain Lab**

```
C:\Users\EXAM\blockchain-toolkit>touch migrations/2_deploy_contracts.js
Touching migrations/2_deploy_contracts.js
```

```
C:\Users\EXAM\blockchain-toolkit>truffle migrate
This version of µWS is not compatible with your Node.js build:

Error: node-loader:
Error: The specified module could not be found.
C:\Users\EXAM\AppData\Roaming\npm\node_modules\truffle\node_modules\ganache\dist\node/1RGFZdPM.node
Falling back to a NodeJS implementation; performance may be degraded.



Compiling your contracts...
===========================
> Compiling .\contracts\MyContract.sol
> Compiling .\contracts\MyContract.sol
> Artifacts written to C:\Users\EXAM\blockchain-toolkit\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)
```

```
2_deploy_contracts.js
=====================

   Deploying 'MyContract'
   ----------------------
   > transaction hash:    0x709d305a57217af56d89eb18e8859d94a527753410a4592066ea33fa090b8d68
   > Blocks: 0            Seconds: 0
   > contract address:    0xA7869863492054ec73F6D89F9d66ED5d4054fd92
   > block number:        1
   > block timestamp:     1666945381
   > account:             0x32DF81A5A4ae372ae5fa7ABDa3bB0D4B10068643
   > balance:             99.99529368
   > gas used:            235316 (0x39734)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00470632 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00470632 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.00470632 ETH
```

```
truffle(development)> app.set('Drishti')
{
  tx: '0x74d7231d677d61e102eb136029daffdad2a4e023ba029209751e689717961807',
  receipt: {
    transactionHash: '0x74d7231d677d61e102eb136029daffdad2a4e023ba029209751e689717961807',
    transactionIndex: 0,
    blockHash: '0xad5edc6bc59eda71701d6fb6aff62a60990ba0245ebb34599a45215e44318382',
    blockNumber: 3,
    from: '0x32df81a5a4ae372ae5fa7abda3bb0d4b10068643',
    to: '0xa7869863492054ec73f6d89f9d66ed5d4054fd92',
    gasUsed: 29030,
    cumulativeGasUsed: 29030,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000',
    rawLogs: []
  },
  logs: []
}
truffle(development)> app.get()
'Drishti'
```

**2. Create a Smart contract to simulate function overloading . Execute the contract using truffle framework.**

**→ touch ./contracts/Overloading.sol**

```
C:\Users\drish\blockchain-toolkit>touch ./contracts/Overloading.sol
Touching ./contracts/Overloading.sol
```

**Code:**

```
pragma solidity >=0.4.2 <=0.8.0;
contract Overloading {
function getCal(uint a, uint b,uint c) public pure returns(uint){
return a*b*c;
}
function getCal(uint a, uint b) public pure returns(uint){
return a + b;
}
function callValueWithTwoArguments() public pure returns(uint){
return getCal(1,2);
}
function callValueWithThreeArguments() public pure
```

```
returns(uint){
return getCal(1,2,3);
}
}
```

**→truffle compile**

```
C:\Users\drish\blockchain-toolkit>truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\MyContract.sol
> Compiling .\contracts\MyContract.sol
> Compiling .\contracts\Overloading.sol
> Artifacts written to C:\Users\drish\blockchain-toolkit\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang

C:\Users\drish\blockchain-toolkit>truffle migrate

Compiling your contracts...
===========================
> Compiling .\contracts\MyContract.sol
> Compiling .\contracts\MyContract.sol
> Compiling .\contracts\Overloading.sol
> Artifacts written to C:\Users\drish\blockchain-toolkit\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)
```

**Name: Drishti Bhatia** **Roll Number: 07**
**SYMCA** **Block Chain Lab**

**→truffle migrate**

```
C:\Users\drish\blockchain-toolkit>truffle migrate

Compiling your contracts...
===========================
> Compiling .\contracts\MyContract.sol
> Compiling .\contracts\MyContract.sol
> Compiling .\contracts\Overloading.sol
> Artifacts written to C:\Users\drish\blockchain-toolkit\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
```

```
Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


2_deploy_contracts.js
=====================

   Replacing 'Overloading'
   -----------------------
   > transaction hash:    0xc3e4d29e386172908f027ab701cb3c57b1eb6efc34c47d35053ec9624f99b9da
   > Blocks: 0            Seconds: 0
   > contract address:    0xa863EEf78f5F7F8613fC95167163289147CFbE4e
   > block number:        2
   > block timestamp:     1667130925
   > account:             0x6a5281aaeBF8D1729C08607cE7D8c6fA7E4F41EF
   > balance:             99.99555084
   > gas used:            110905 (0x1b139)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.0022181 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:           0.0022181 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.0022181 ETH
```

**→truffle console**

**→ Overloading.deployed().then((instance) => {app = instance} )**

```
C:\Users\drish\blockchain-toolkit>truffle console
truffle(development)> Overloading.deployed().then((instance) => {app = instance} )
undefined
```

**Name: Drishti Bhatia** **Roll Number: 07**
**SYMCA** **Block Chain Lab**

**→app.getCal(8,8)**
**→app.callValueWithTwoArguments()**
**→callValueWithThreeArguments ()**

```
truffle(development)> app.getCal(8,8)
BN { negative: 0, words: [ 16, <1 empty item> ], length: 1, red: null }
truffle(development)> app.callValueWithTwoArguments()
BN { negative: 0, words: [ 3, <1 empty item> ], length: 1, red: null }
truffle(development)> app.callValueWithThreeArguments()
BN { negative: 0, words: [ 6, <1 empty item> ], length: 1, red: null }
truffle(development)>
```

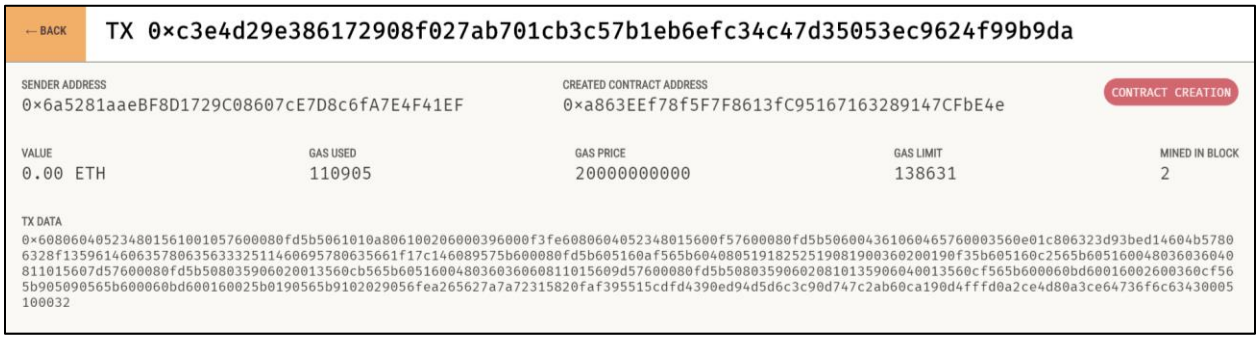
**→ const inst8=await Overloading.deployed()**
**→inst8.methods['getSum(uint256,uint256,uint256)'](8,8,8)**

```
truffle(development)> const inst_prac8=await Overloading.deployed()
undefined
truffle(development)> const inst8=await Overloading.deployed()
undefined
truffle(development)> inst8.methods['getCal(uint256,uint256,uint256)'](8,8,8)
BN {
  negative: 0,
  words: [ 512, <1 empty item> ],
  length: 1,
  red: null
}
```

**→inst8.methods['getSum(uint256,uint256)'](8,8)**

```
truffle(development)> inst8.methods['getCal(uint256,uint256)'](8,8)
BN { negative: 0, words: [ 16, <1 empty item> ], length: 1, red: null }
```

**Transaction Loss:**




**Conclusion:** We have successfully executed the smart contract using the truffle framework.