

Blockchain and Bitcoin

Blockchain Concepts

- Introduction to DLT and Blockchain
- Double Spending problem
- Distributed, Decentralized systems
- CAP Theorem
- A deep dive into Blockchain Architecture
 - 1) Hashing or digital finger print
 - 2) Hashing in real time, types of hashing (Ex : Merkle Tree)
 - 3) Cryptography, types of cryptography
 - 4) Cryptography in block chain and digital signature
 - 5) Transactions and Authorizing a transaction
 - 6) Storing transaction data
 - 7) Protecting data store
 - 8) Hash puzzles, nonce, proof - of - work, verifying transactions
 - 9) Distributing data store among peers
 - 10) Mining, rewards and punishment
 - 11) Choosing transaction history
 - 12) What is 51 % CPU attack ?
- Different Blockchain Platforms

Bitcoin

- Introduction to Bitcoin and History of Bitcoin
- Why, Where and How to buy Bitcoins?
- Wallets and Exchanges
- Bitcoin Transaction Flow
- Consensus in Bitcoin and Forking

Ethereum & Solidity

Ethereum

- Introduction to Ethereum
- External accounts and contract accounts
- What is a smart contract ?
- EVM, Ether, Gas
- Meta mask, Remix
- Installing and creating an Ethereum wallet in meta mask.
- Choosing a chain for your development environment

Blockchain

- Get ethers from Ropsten faucet
- Write "Hello World" - basic smart contract
- Write smart contracts and execute in Ropsten EVM
- Geth client set up, Geth CLI
- Setting a Private chain using Geth
- Installation and configuration of the Ethereum client Parity
- Test RPC Configuration and usage
- BaaS(Blockchain as a Service)
- Infura - Scalable Blockchain Infrastructure
- ERC Standards: ERC20, ERC223, ERC721
- Casper Introduction

Solidity - Programming language to write smart contracts

- Solidity contract layout
- Data types
- Arrays, struct, mappings, event
- Functions, Function visibility
- Fallback functions and payable
- Custom modifiers
- Inheritance, Interfaces
- Error handling using - assert, revert, require
- Creating and deploying tokens
- Time based events
- Develop smart contracts using solidity

web3.Js & Truffle Framework

- Installing and Running web3.js
- API Reference
 - 1) Web3
 - 2) web3.eth
 - 3) web3.eth.Contract
 - 4) web3.eth.accounts
- Truffle Commands
- Truffle boxes, Creating a Project
- Compiling, Interacting, Debugging Your contracts
- Running Migrations, Truffle with MetaMask
- Testing your Contracts

Project Case Study

- ICO(Initial Coin Offering)
- KYC(Know your Customer)

Hyperledger

- Introduction to Hyperledger Project
- Modules and Frameworks in Hyperledger

Hyperledger Fabric

- Introduction to Hyperledger Fabric
- How Hyperledger Fabric works ? Basic workflow transaction
- Architecture
- Main components of Hyperledger
 - 1) Fabric CA
 - 2) Fabric Peer
 - 3) Fabric Ordering service
- Channels
- Chain codes
- Membership Service Providers (MSP)
- How to create your own Hyperledger Fabric network?
- How to configure Peers?
- How to create channels?
- How to join peers to multiple channels?
- How to write and instantiate Chain code?

Hyperledger Composer

- Introduction to Hyperledger Composer
- Architecture
- Modeling Language
- Access Control Language
- Transaction Processor Functions
- Query Language
- Setting up the Development environment
- Developing Business Networks
 - 1) Business Network Definitions
 - 2) Create a Business Network Definition
 - 3) Deploying Business Networks
 - 4) Testing Business Networks
 - 5) Publish Business Network Definitions
 - 6) Upgrading a deployed Business network
- Developing Applications
 - 1) Writing a Node.js application
 - 2) Writing Web applications

Project Case Study

- Letter Of Credit
- supply chain model

Blockchain Research

- Blockchain and AI
- Blockchain and Big data
- Blockchain and SAP
- Quorum
- Corda
- Ripple
- Oraclize
- Atomic swaps
- STORJ
- IPFS

Prerequisites

- Knowledge on any Programming Language(Compulsory)
- Full stack development Knowledge is an added advantage