Block Chain Application

**Dependencies :**

Node

Truffle

Ganashe ( Local Block Chain )

Metamask

Ganashe – Provided 10 account with unique ids this will represent the Voters in the elections app

Testing the Application: ( Via Mocha Testing Framework and Chai )

**Commands**:

(Asynchronous call )

Election.deployed().**then**(function(instance){**app** = instance})

Defining the instance as an **app** variable

Truffle console ( Enter Into Truffle console )

web3.eth.accounts ( List All accounts )

Solidity Basics:

**Data Types**

Integer : 1. Uint ( unsigned ) VS int ( signed )

**Address** ( send VS transfer )

String ( uses a lot of gas – instead use bits

Enumerator ( custom data types )

**Data Structures**

Static and Dynamic

**Visibility Specifiers**

Private – only visible in current contract

Public – Visible externally and internally ( creates a getter function for storage / state variables )

Internal – Only visible internally

External – only visible externally ( only for functions )

**Function modifiers ( used to change the behavior of functions )**

**Pure** – Cannot read or modify state of function

**View** - Cannot modify state of function

**Constant** – Only for state variables, disallows engagement

**Payable** – Allows function to receive ether

**Error Handling ( state – reverting exceptions )**

Assert() – check internal errors / check invariants

Require() – used for ensuring valid conditions / return calls

Revert() – revert current call on certain condition

Throw () – Do not use ( Deprecated )

**Interview Questions:**

<https://www.youtube.com/watch?v=XIDlYm11lCY>

**Testing**

There are some packages that help check code style, small security issues, and test coverage (Solhint, Solium, Solidity coverage).