BlockChain:

**Block**

* int Height
* Int64 TimeStamp
* byte[] PrevHash
* byte[] Hash
* Transaction[] Transactions
* string Creator

**Transaction:**

* long TimeStamp
* string Sender
* string Recipient
* double Amount
* double Fee

Create a List of transaction (Transaction Pool to hold the complete data temp)

**Create Block**:

* Height = sequence no of a block (An incremental value)
* Timestamp
* Hash (Hash Creation step is below)
* PrevHash will be taken from prev hash (From list of block check last list hash by count-1)
* Transaction Array
* Add the block to block list

**Process to create Hash**:

* Convert timestamp to byte
* Convert the transaction array to byte and its name will be **transactionHash**
* Get the previous hash
* Create a new byte array and keep its length (timeStamp.Length + PrevHash.Length + transactionHash.Length)
* copy bytes of timestamp, transactions and previous hash to a single byte array called headerBytes
* now generate the hash using sha.computehash

Note: if there is no transactions (you are creating for the first time then you will add a block with genesis account which will be utilize when a new user want to have amount in his wallet)

* Amount = 10000000,
* Sender = "System",
* Recipient = "Genesis Account",
* Fee = 0.0001