**Lecture 9 Notes**

Slide 4

* A smart contract is a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein exist across a distributed, decentralized blockchain network. The code controls the execution, and transactions are trackable and irreversible.
* Smart contracts permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a central authority, legal system, or external enforcement mechanism.
* While blockchain technology has come to be thought of primarily as the foundation for bitcoin​, it has evolved far beyond underpinning the virtual currency.
* *Smart contracts are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code.*
* *Nick Szabo, an American computer scientist who invented a virtual currency called "Bit Gold" in 1998, defined smart contracts as computerized transaction protocols that execute terms of a contract.*
* *Smart contracts render transactions traceable, transparent, and irreversible.*

*Slide 6*

* Blockchain technology has become the jet fuel for propelling the concept of the smart contract in the world of business transactions.
* Large corporate institutions are rapidly adopting this concept to remain ahead of their competitors.
* This has led to a call for an industry-wide initiative for formalizing the design framework of a smart contract.
* One of the big four auditing firms, Deloitte, teamed up with the advisory board of the Digital Chambers of Commerce, has come up with a six-point anatomy to formalize a smart contract design

Slide 6

* **Autonomy –** You’re the one making the agreement; there’s no need to rely on a broker, lawyer or other intermediaries to confirm. Incidentally, this also knocks out the danger of manipulation by a third party, since execution is managed automatically by the network, rather than by one or more, possibly biased, individuals who may err.
* **Trust** – Your documents are encrypted on a shared ledger.  There is no way that someone can say they lost it.
* **Backup** – Imagine if your bank lost your savings account. On the blockchain, each one of your friends have your back. Your documents are duplicated many times over.
* **Safety** – Cryptography, the encryption of websites, keeps your documents safe. There is no hacking. In fact, it would take an abnormally smart hacker to crack the code and infiltrate.
* **Speed** – You would ordinarily have to spend chunks of time and paperwork to manually process documents. Smart contracts use software code to automate tasks, thereby shaving hours off a range of business processes.
* **Savings** – Smart contracts save you money since they knock out the presence of an intermediary. You would, for instance, must pay a notary to witness your transaction.
* **Accuracy** – Automated contracts are not only faster and cheaper but also avoid the errors that come from manually filling out heaps of forms.

Slide 8

* **Smart contract code**: For instance, Ethereum Solidity code (that is stored, verified and executed on a blockchain).
* **Smart legal contracts**: These are written as a specification for using smart contract code as a complement/substitute for legal contracts executed in traditional usage.

Slide 9

* This statement defines that the smart contract (source file) will not compile with a compiler earlier than 0.4.0 and the version after 0.6.0. By introducing this declaration, no unintended behaviour will be introduced if a new version of the compiler is introduced.
* *There is a standard structure followed by solidity smart contracts. Any smart contract starts with the following statement.*

Slide 10

* It is an object-oriented high-level programming language used to implement smart contracts.
* Solidity helps create contracts for crowdfunding, blind auctions, voting, and multi-signature wallets.
* Solidity is influenced by languages such as Python, JavaScript, and C++.

Slide 11

* There are two types of variables one needs to be familiar with in smart contracts.
* **Value Type** – These variables are passed by value. This means that they are always copied when they are used in assignments or as function arguments. Integers and booleans addresses are some prominent examples.
* **Reference Type** – These variables are of complex types and are passed by reference. Copying these is expensive, and hence it needs to be managed carefully, and these variables do not fit into 256 bit.

Slide 12

**Contract Declaration**

* + This is declared through the keyword ‘contract.’
  + This will declare an empty contract identified by the name of “PurchaseOrder.”

This is depicted as follows:

 contract PurchaseOrder{

 }

Slide 13

* INT -refers to an integer.
* U -means unsigned, which means that this type can represent only positive integers and not both positive and negative integers.
* 25 -This refers to the 256 bits in size.
* The minimum value of uint 256 can be assigned to- 0.
* The maximum value that can be assigned for uint 256 is- 2^256-1.

Slide 14

* A constructor is one that is called at the time of deployment of a contract.
* The constructor uses some values to initialize the contract.
* It is also possible to create a parameterized constructor which can be created by passing a variable and initializing the function using the passed in value.
* The access modifier “public” associated with the constructor is the key point to note here.
* The keyword “public” denotes that anyone can access the function. Hence, this is not a restricted function.

Slide 15

* Functions refer to controlled capabilities that can be added to a program.
* Any function will always be preceded by the keyword function.
* **A function declaration looks like this:** 
  + - **“function <function name> <access modified> <state mutator> <return value>”**

Slide 16

* The most common requirement is to read the stored value.
* For example, let us consider the product\_quantity value.
* To provide this capability, a read or get function is added.
* In this function, we will not manipulate the stored value, but will just retrieve the stored value.
* The get function can be broken down as follows:
  + - **Keyword-Function name; Value- get\_quantity()**
    - **Keyword- Access modifier; Value- Public (The function can be accessed by anyone)**
    - **Keyword- state-mutator; Value- View (The function only reads the state of the contract and does not change it)**
    - **Keyword- Returns; Value- Defines a variable of type uint256.**

### Slide 17

### Setter Function

* It is necessary to read the data, but it is also necessary to have the capability to write/update data.
* This capability is added through a setter function.
* This function will take a value from the user in the form of an input parameter.
* We will now break down our set function. We will now add a sample function to update the value of product quantity.
* **Keyword- Function name; Value- update\_quantity**
* **Keyword- Access modifier; Value- Public**
* **Keyword- state-mutator; Value- This is not needed as the state is updated by the functions.**
* **Keyword- Returns; Value- Returns a variable of type uint256.**

Slide 18

* It is now time to deploy the smart contract.
* For example: Remix Online IDE to test the smart contract.
  + - Remix is an online playground that is used for Ethereum smart contracts.
    - It is completely based on a browser.
    - It provides an integrated development environment where you can write your smart contracts.
    - Remix provides an online solidity compiler capability.
    - Remix IDE helps compile a smart contract seamlessly using a specific compiler version.
    - It helps in the quick testing of a smart contract.
* **Which of the following is NOT something you get by using smart contracts:**
* Trust
* Safety via cryptography
* Cost savings
* Human Error
* **When an event outlined in the smart contract is triggered, for instance price change or expiration of a contract, then:**
* the code executes
* the escrow amount is released
* the contract expires
* none of the above
* **A smart contract allows you to virtually exchange anything (money, shares, real estate, etc. ) anything without:**
* Additional cost
* Services of middlemen
* Transparency
* Blockchain