SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

COURSE NAME: 19OE201-Blockchain Technology

IV YEAR /VII SEMESTER

Unit 3- ETHEREUM

Topic : Solidity Programming Language: Functions

Solidity

Solidity functions are self-contained modules of code that accomplish a specific task.

Solidity allows the developer to write modular code by using functions to eliminate the redundancy of rewriting the same piece of code.

Solidity functions are the building blocks of Ethereum smart contracts, allowing developers to define the logic that executes on the blockchain.

They play a crucial role in interacting with the Ethereum network, processing transactions, and managing the state of decentralized applications (DApps).

Solidity Syntax

Pragma: In Solidity, a pragma language will specify how the compiler will process any type of input. Typically, the first line of code in Solidity based smart contracts contains the pragma.

Contract: uintstoredData component within the code will denote the Solidity contract. This part will contain all the data and code needed for locating a particular address within the blockchain.

File Importing: Solidity offers similar support for file import systems like JavaScript.

Functions

The most common way to define a function in Solidity is by using the **function** keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

```
function function-name(parameter-list) scope returns()
 {
//statements
}
```

Functions

```
pragma solidity ^0.5.0;

contract Test {

function getResult() public view returns(uint)
{
uint a = 1; // local variable
uint b = 2;
uint result = a + b;
return result;
}
}
```

The return Statement

Function Return Variables:

- Return variables in Solidity functions serve the crucial purpose of conveying information back to the caller.
- They are declared using the `returns` keyword and can be either named or unnamed.

```
pragma solidity ^0.5.0;
 contract Test {
function getResult()
 public view returns(uint product, uint sum)
{
uint a = 1; // local variable
 uint b = 2;
 product = a * b;
sum = a + b; //alternative return statement to return //multiple values //
return(a*b, a+b);
 }
 }
```

The return Statement

Named Return Variables

- When you define named return variables, you make it explicit what the function is returning.
- This clarity is especially valuable in complex smart contracts.

Unnamed Return Variables

- Unnamed return variables can be useful in situations where the return value is straightforward and doesn't require an explicit name.
- They are commonly used for functions with a single return value, such as getter functions for state variables.

References

TEXT BOOKS

1. Mastering Bitcoin: Unlocking Digital Cryptocurrencies, by Andreas M Antonopoulos 2018
2. Imran Bashir, “Mastering Blockchain: Distributed Ledger Technology, Decentralization and Smart Contracts Explained”, Second Edition, Packt Publishing, 2018.
3. https://101blockchains.com/blockchain-vs-database-the-difference/

REFERENCES

1. William Mougayar, “Business Blockchain Promise, Practice and Application of the Next Internet Technology, John Wiley & Sons 2016.
2. Josh Thompson, ‘Blockchain: The Blockchain for Beginnings, Guild to Blockchain Technology and Blockchain Programming’, Create Space Independent Publishing Platform, 2017.
3. Arvind Narayanan, “Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction”, Princeton University Press, July 19, 2016.
4. Henning Diedrich, Ethereum: Block chains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations-2016

Thank You