ASSIGNMENT 07

1)If local variables are not stored in contract storage, then how do they become available during execution?

Ans: Stack hold is used to hold small local variables. It can only hold a limited number of values. Local variables are defined inside the function and kept in stack location. During different function calls the local variables do not save their value and they stay till the function is concluded.

2)Difference between non-payable and payable functions?

Ans:1. function funcA()public{} And function funcB()public payable{}

2. we can use the keyword payable to specify that an address or a function payable.

3. Can receive Ether.

Non-Payable Function:

1.Syntax: function funcA() public{} we can use the keyword payable to specify that an address or a function is payable.

2.No keyword is used.

3.Cannot receive Ether.

3)Why does Solidity have only One Constructor?

Ans: Solidity provides a constructor declaration inside the smart contract and it invokes only once when the contract is deployed and is used to initialize the contract state. A default constructor is created by the compiler if there is no explicitly defined constructor.

4) What is the Difference between tx.origin & msg.sender?

Ans:

With msg.sender the owner can be a contract and with tx.origin the owner can never be a contract.

msg.sender gives the direct sender of the message, so for example a contract that passed it along. tx.origin gives the origin of he transactions, so the user address it was originally sent from. In practice this will always be a user so eth's answer holds true.

msg.sender: Refers to the address of an account or a smart contract which is directly calling a smart contract's function.

tx.origin: Refers to the address of an account which is calling a smart contract's function, only account address can be tx.origin.