

## Day 5

### Advance Blockchain

#### Answer 1:

Solidity has many functions for error handling. Errors can occur at compile time or runtime. Solidity is compiled to byte code and there a syntax error check happens at compile-time, while runtime errors are difficult to catch and occurs mainly while executing the contracts. Some of the runtime errors are out-of-gas error, data type overflow error, divide by zero error, array-out-of-index error, etc. Until version 4.10 a single throw statement was there in solidity to handle errors, so to handle errors multiple if...else statements, one has to implement for checking the values and throw errors which consume more gas. After version 4.10 new error handling construct assert, require, revert statements were introduced, and the throw was made absolute.

#### Require Statements

The 'require' statements declare prerequisites for running the function i.e. it declares the constraints which should be satisfied before executing the code. It accepts a single argument and returns a boolean value after evaluation, it also has a custom string message option. If false then exception is raised and execution is terminated. The unused gas is returned back to the caller and the state is reversed to its original state. Following are some cases when require type of exception is triggered :

- When require() is called with the arguments which result as false.
- When a function called by a message does not end properly.
- When a contract is created using the new keyword and the process does not end properly.
- When a codeless contract is targeted to an external function.
- When ethers are sent to the contract using the public getter method.
- When .transfer() method fails.
  - When an assert is called with a condition that results in false.
  - When a zero-initialized variable of a function is called.
  - When a large or a negative value is converted to an enum.
  - When a value is divided or modulo by zero.
  - When accessing an array in an index which is too big or negative.

## Assert Statement

Its syntax is similar to the require statement. It returns a boolean value after the evaluation of the condition. Based on the return value either the program will continue its execution or it will throw an exception. Instead of returning the unused gas, the assert statement consumes the entire gas supply and the state is then reversed to the original state. Assert is used to check the current state and function conditions before the execution of the contract. Below are some cases with assert type exceptions :

- When an assert is called with a condition that results in false.
- When a zero-initialized variable of a function is called.
- When a large or a negative value is converted to an enum.
- When a value is divided or modulo by zero.
- When accessing an array in an index which is too big or negative.

## Answer 2:

```
pragma solidity >=0.5.13 <0.7.3;
```

```
// SPDX-License-Identifier: MIT
```

```
contract ourBank {
```

```
    function deposit() public payable {  
    }
```

```
    function withdraw(uint amount) public {  
        address payable to = msg.sender;  
        to.transfer(amount);  
    }
```

```
    function balance() public view returns(uint) {  
        return address(this).balance;  
    }
```

```
}
```