SOLIDITY -CONTRACT

Contract in Solidity is similar to a Class in C++. A Contract have following properties.

Constructor − A special function declared with constructor keyword which will be executed once per contract and is invoked when a contract is created.

State Variables − Variables per Contract to store the state of the contract.

Functions − Functions per Contract which can modify the state variables to alter the state of a contract.

Visibility Quantifiers

Following are various visibility quantifiers for functions/state variables of a contract.

-

external − External functions are meant to be called by other contracts. They cannot be used for internal call. To call external function within contract this.function_name() call is required. State variables cannot be marked as external.

-
-

public − Public functions/ Variables can be used both externally and internally. For public state variable, Solidity automatically creates a getter function.

-
-

internal − Internal functions/ Variables can only be used internally or by derived contracts.

-
-

private − Private functions/ Variables can only be used internally and not even by derived contracts.

-

Example

```
pragma solidity ^0.5.0;


contract C {
  //private state variable
  uint private data;

  //public state variable
  uint public info;

  //constructor
  constructor() public {
    info = 10;
  }
  //private function
  function increment(uint a) private pure returns(uint) { return a + 1; }

  //public function
  function updateData(uint a) public { data = a; }
  function getData() public view returns(uint) { return data; }
  function compute(uint a, uint b) internal pure returns (uint) { return a +
b; }}//External Contract
contract D {
  function readData() public returns(uint) {
    C c = new C();
    c.updateData(7);
    return c.getData();
  }}//Derived Contract
contract E is C {
  uint private result;
  C private c;

  constructor() public {
    c = new C();
  }
  function getComputedResult() public {
    result = compute(3, 5);
```

```
 }
function getResult() public view returns(uint) { return result; }
function getData() public view returns(uint) { return c.info(); }}
```

Run the above program using steps provided in Solidity First Application chapter. Run various method of Contracts. For E.getComputedResult() followed by E.getResult() shows −

Output

```
0: uint256: 8
```