Fall 2024

SEC-205: Distributed Ledger and Blockchain

Lab/Practical Session 2: Data Structure in Solidity

Charnon Pattiyanon, Ph.D.

charnon@cmkl.ac.th

Assistant Director of IT and Instructor

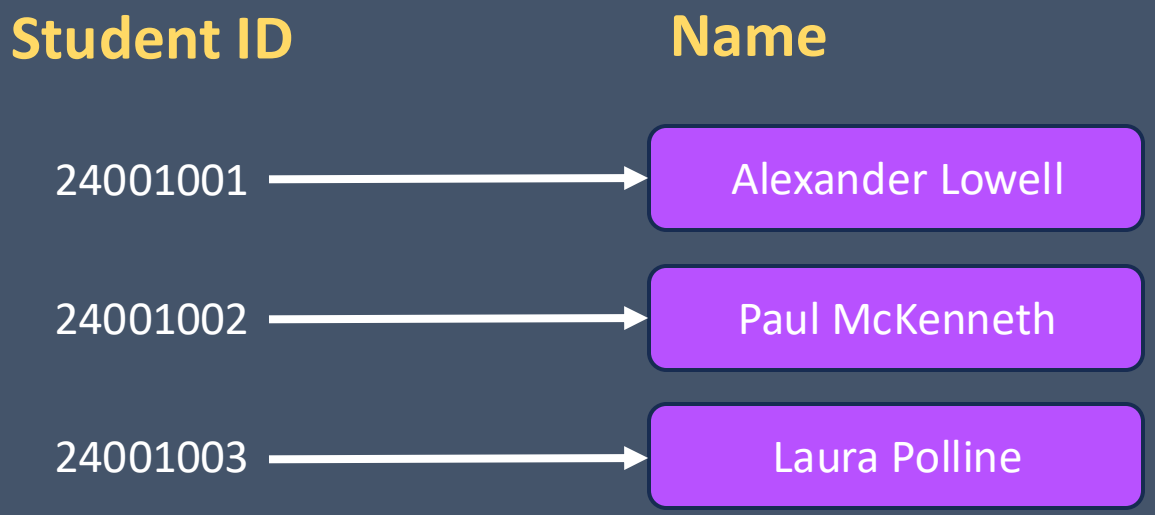
Department of Artificial Intelligence and Computer Engineering (AiCE)

CMKL University

Basic Twitter (X) Smart Contracts

Mapping in Solidity

- It is a pair between a key and value



Mapping in Solidity

- It is a pair between a key and value

Address	Name
0x3794D3…5FB6	Alexander Lowell
0xB67B07…B490	Paul McKenneth
0xD7644B…3504	Laura Polline

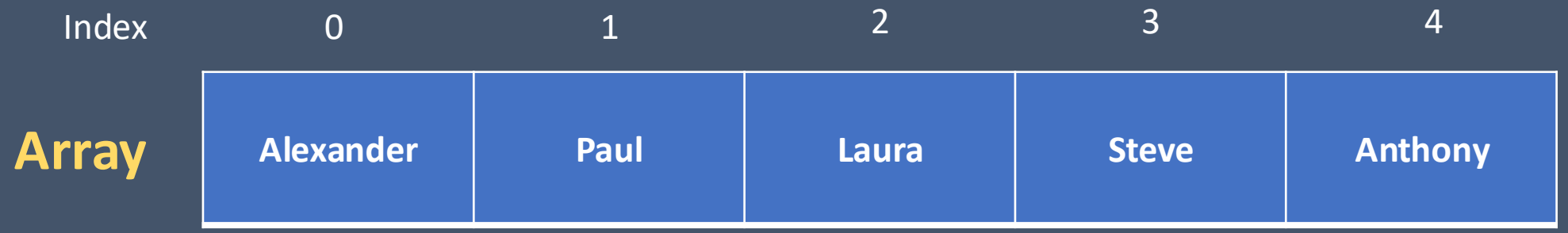
Mapping in Solidity

```solidity
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.26

contract Mapping {

  // Mapping from address to uint
  mapping (address => uint8) public myMap;

  function get(address _addr) public view returns (uint8) {
    // Mapping always returns a value
    // If the value was never set, it will return the default value
    return myMap[_addr];
  }

  function set(address _addr, uint8 _i) public {
    // Update the value at this addresss
    myMap[_addr] = _i;
  }

  function remove(address _addr) public {
    // Reset the value to the default value
    delete myMap[_addr];
  }
}
```

Complete Our Basic Twitter Smart Contract

Arrays in Solidity

- An array is simply a list of things, where each index points to a value.



Arrays in Solidity

- An array is simply a list of things, where each index points to a value.

```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.26

contract Array {
  // There are several ways to initialize an array
  uint256[] public arr;
  uint256[] public arr2 = [1, 2, 3];

  // Fixed size array, all elements initialize to 0
  uint8[10] public myFixedArry;
  string[5] public names;

  function get(uint256 i) public view returns (uint256) {
    return arr[i];
  }

  function pushToDynamicArr(uint256 value) public {
    arr.push(value);
  }

  function getDynamicArrLength() public view returns (uint256){
    return arr.length;
  }

}
```

Add Arrays into Our Twitter Smart Contract

Structs in Solidity

- Struct is a syntax in Solidity to describe the structure of something.

Model	Y
Year	MY2024
Color	Blue
Configuration	Performance AWD
Price	2,329,000 THB

Structs in Solidity

- There are two steps to define and use structs in Solidity

1 Define a struct

```sol
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.26

contract TeslaRegistry {

  struct Tesla {
    string model;
    uint16 year;
    string color;
    uint128 mileage;
    string vin;
    string driving_mode;
    uint128 price;
  }

  Tesla[] public teslas;
}
```

2 Add data

```sol
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.26

contract TeslaRegistry {
  //...
  function addTestla(string memory model, uint16 memory year, string
memory color, uint128 memory mileage, string memory vin, string memory
driving_mode, uint128 memory price) public {
    Tesla memory newTesla = Tesla({
      model: model,
      year: year,
      color: color,
      mileage: mileage,
      vin: vin,
      driving_mode: driving_mode,
      price: price
    });

    teslas.push(newTesla);
  }
}
```

Add Structs into Our Twitter Smart Contract

Require Statement in Solidity

- **Require** is a syntax to make sure that the defined conditions are true.

IF Conditions are true **{**

Continue

}

Otherwise {

Revert the transaction

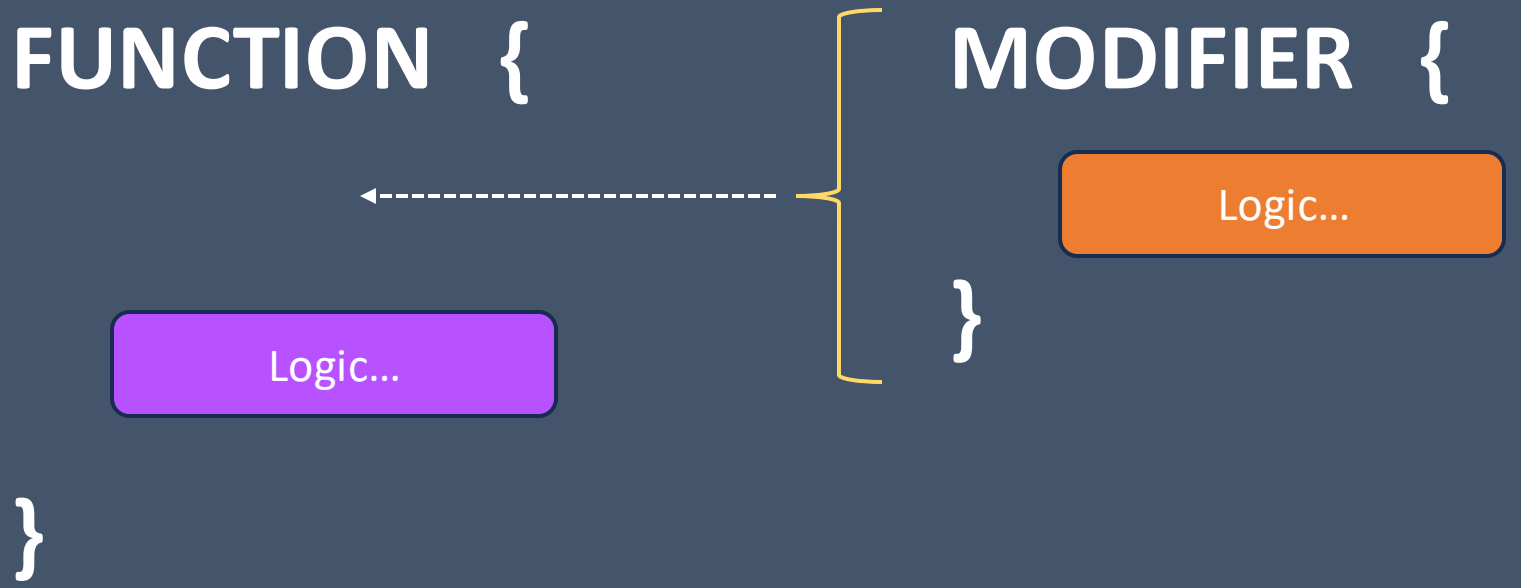
}

```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.26

contract bigCPromotion {
  uint16 public maxSaleItems = 10;

  function checkout(uint16 amount) public {
    require(amount <= maxSaleItems, "No more than 10 items allowed.");

    ...
    _checkout(amount);
  }
}
```

Use Require to limit tweets' length in Our Twitter Smart Contract

Modifiers in Solidity

- Modifier is a syntax to add behaviors to a function



Modifiers in Solidity

- Make sure that only the OWNER can call the changeOwner function

```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.26

contract Test {
  address public owner;

  constructor(){
    owner = msg.sender;
  }

  modifier onlyOwner() {
    require(msg.sender == owner, "You are not the owner.");
    _;
  }

  function changeOwner(address newOwner) public onlyOwner {
    owner = newOwner;
  }
}
```

Add Custom Modifiers to our Twitter Smart Contract

End of the Lab! 

Please feel free to ask any questions.

If you need further discussion, please contact me:

- Email me at charnon@cmkl.ac.th
- Appoint me for 1-on-1 discussion during the office hours.