Smart Contract

```
pragma solidity >0.4.0 <0.6.0;

contract Contract_A {

    uint256 value;
    constructor (uint256 n) public {
        value = n;
    }

function incrementValue(uint256 x) public {

        value += x;
    }
}

contract Contract_B is Contract_A(22){

    function getValue() public view returns(uint256) {
        return value;
    }
}
```

B.2 Multilevel Inheritance

Smart Contract

```
pragma solidity >0.4.0 <0.6.0;

contract Contract_A {

    uint256 value;
    constructor (uint256 n) public {
        value = n;
    }

    function incrementValue(uint256 x) public {
        value += x;
    }
}

contract Contract_B is Contract_A(22) {

    function decrementValue(uint256 x) public {
        value -= x;
    }
}

contract Contract_C is Contract_B {

    function getValue() public view returns(uint256) {
        return value;
    }
}
```

B.3 Hierarchical Inheritance

Smart Contract

```
pragma solidity >0.4.0 <0.6.0;

contract Contract_A{

    uint256 value;
```

```
    constructor (uint256 n) public{
        value = n;
    }

    function getValue() public view returns(uint256){
        return value;
    }
}

contract Contract_B is Contract_A(22){
    function incrementValue(uint256 x) public {
        value += x;
    }
}

contract Contract_C is Contract_A(25){
    function decrementValue(uint256 x) public {
        value -= x;
    }
}
```

B.4 Multiple Inheritance

Smart Contract

```
pragma solidity >0.4.0 <0.6.0;

contract Contract_A{

    uint256 value;
    constructor (uint256 n) public{
        value = n;
    }

    function incrementValue(uint256 x) public {
        value += x;
    }
}

contract Contract_B is Contract_A{
    function decrementValue(uint256 x) public {
        value -= x;
    }
}

contract Contract_C is Contract_A(25), Contract_B{
    function getValue() public view returns(uint256){
        return value;
    }
}
```
