

Create a .sol file on REMIX with:

```
pragma solidity ^0.4.18;
```

```
contract Courses {
```

```
    struct Instructor {  
        uint age;  
        string fName;  
        string lName;  
    }
```

```
    // Mapping Instructor struct to an Ethereum address.
```

```
    mapping (address => Instructor) instructors;
```

```
    address[] public instructorAccts;
```

```
    function setInstructor(address _address, uint _age, string _fName, string _lName) public {  
        var instructor = instructors[_address];
```

```
        instructor.age = _age;  
        instructor.fName = _fName;  
        instructor.lName = _lName;
```

```
        instructorAccts.push(_address) -1;  
    }
```

```
    // Return a list of addresses from instructorAccts
```

```
    function getInstructors() view public returns(address[]) {  
        return instructorAccts;  
    }
```

```
    // Retrieve a specific instructor based on a provided address
```

```
    function getInstructor(address _address) view public returns (uint, string, string) {  
        return (instructors[_address].age, instructors[_address].fName,  
instructors[_address].lName);  
    }
```

```
    // Count how many instructors
```

```
    function countInstructors() view public returns (uint) {  
        return instructorAccts.length;  
    }
```

```
}
```

If you got the error with 'Warning: Use of the "var" keyword is deprecated' when you try to compile the .sol file,

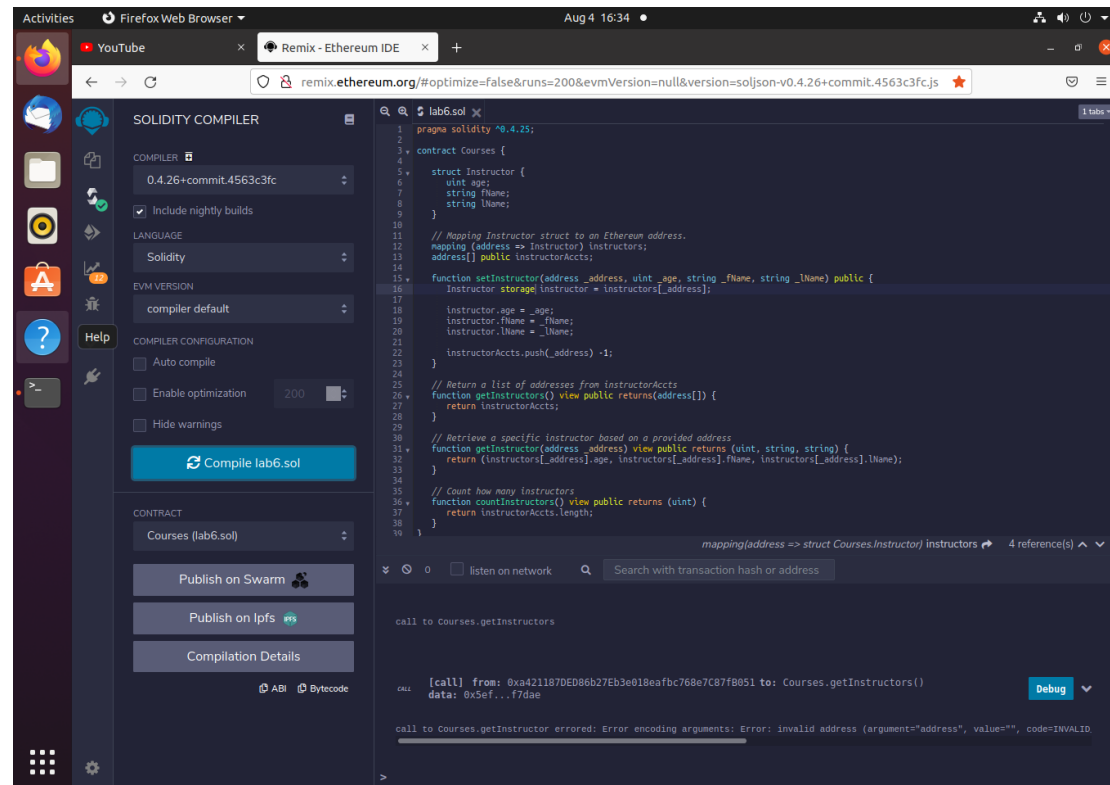
We change the

```
'var instructor = instructors[_address];'
```

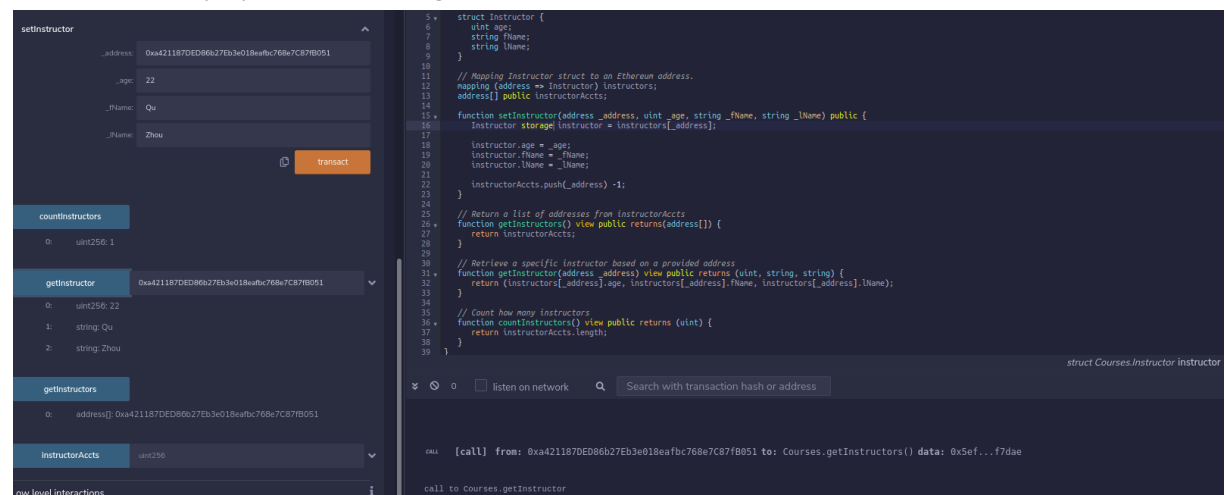
To

```
'Instructor storage instructor = instructors[_address];'
```

Then problem should be fixed



Then we click Deploy to do the testing (address should use the account address):



I enter the second list of data, then click **CountInstructor** it change to 2

```
pragma solidity ^0.4.26;
contract Courses {
    struct Instructor {
        uint age;
        string fName;
        string lName;
    }
    // Mapping Instructor struct to an Ethereum address.
    mapping (address => Instructor) instructors;
    address[] public instructorAccts;
    function setInstructor(address _address, uint _age, string _fName, string _lName) public {
        Instructor storage instructor = instructors[_address];
        instructor.age = _age;
        instructor.fName = _fName;
        instructor.lName = _lName;
        instructorAccts.push(_address);
    }
    // Return a list of addresses from instructorAccts
    function getInstructors() view public returns(address[]) {
        return instructorAccts;
    }
    // Retrieve a specific instructor based on a provided address
    function getInstructor(address _address) view public returns (uint, string, string) {
        return (instructors[_address].age, instructors[_address].fName, instructors[_address].lName);
    }
    // Count how many instructors
    function countInstructors() view public returns (uint) {
        return instructorAccts.length;
    }
}
```

CountInstructor number will increase by how many dataset user entered:

```
setInstructor
_address: 0xa421187DED86b27Eb3e018eafbc768e7C87fB051
_age: 22
_fName: Quuuuuyafa
_lName: faZhafafafafafsa
transact

countInstructors
0: uint256: 6

getInstructor
0xa421187DED86b27Eb3e018eafbc768e7C87fB051
0: uint256: 22
1: string: Quuuu
2: string: Zhou

getInstructors
0: address[]: 0xa421187DED86b27Eb3e018eafbc768e7C87fB051,0xa421187DED86b27Eb3e018eafbc768e7C87fB051
```