pragma solidity ^0.4.18;

contract Owned {

address owner;

function Owned() public {

owner = msg.sender;

}

modifier onlyOwner {

require(msg.sender == owner);

\_;

}

}

contract Courses is Owned {

struct Instructor {

bytes16 aName;

bytes16 crime;

bytes16 proof;

bytes16 astatus;

bytes16 hash;

bytes16 BlockHash;

address acountaddress;

}

mapping (address => Instructor) instructors;

address[] public instructorAccts;

event instructorInfo(

bytes16 aName,

bytes16 proof,

bytes16 crime,

bytes16 hash,

bytes16 BlockHash

);

// event Crimes(

// bytes16 BlockHash,

// bytes16 crime,

// bytes16 proof,

// bytes16 astatus,

// bytes16 hash

//);

function setInstructor(address \_address, bytes16 \_aName, bytes16 \_crime, bytes16 \_proof,bytes16 \_astatus,bytes16 \_hash, bytes16 \_BlockHash) onlyOwner public {

var instructor = instructors[\_address];

instructor.acountaddress = msg.sender;

instructor.aName = \_aName;

instructor.crime = \_crime;

instructor.proof = \_proof;

instructor.astatus = \_astatus;

instructor.hash = \_hash;

instructor.BlockHash = \_BlockHash;

instructorAccts.push(\_address) -1;

instructorInfo(\_aName,\_proof, \_crime, \_hash,\_BlockHash);

}

//function updateDetails(address \_address,bytes16 \_BlockHash,bytes16 \_crime,bytes16 \_proof,bytes16 \_astatus,bytes16 \_hash) onlyOwner public {

// var instructor = instructors[\_address];

//instructor.BlockHash = \_BlockHash;

//instructor.crime = \_crime;

//instructor.proof = \_proof;

//instructor.astatus = \_astatus;

//instructor.hash = \_hash;

//instructorAccts.push(\_address) -1;

// Crimes(\_BlockHash,\_crime,\_proof, \_astatus, \_hash);

// }

function getInstructors() view public returns(address[]) {

return instructorAccts;

}

function getInstructor(address \_address) view public returns (bytes16,bytes16,bytes16, bytes16) {

return ( instructors[\_address].crime, instructors[\_address].proof,instructors[\_address].astatus,instructors[\_address].hash);

}

function countInstructors() view public returns (uint) {

return instructorAccts.length;

}

}