#1 Contract

pragma solidity ^0.4.2;

contract Election {

//Store candidate

//Read candidate

string public candidate; //declare variable

//Constructor

function Election () public {

candidate = "Niharika"; //state variable (without underscore)

}

}

#2 Contract

pragma solidity ^0.4.2;

contract Election {

//Model candidate

struct Candidate {

uint id;

string name;

uint voteCount;

}

//Store candidates

mapping(uint => Candidate) public candidates; //Key Value pair

//Fetch candidates

//Store candidate count

uint public candidatesCount;

function Election() public {

addCandidate("Niharika");

addCandidate("Rachel");

addCandidate("Tom");

}

function addCandidate (string \_name) private {

candidatesCount++;

candidates[candidatesCount] = Candidate(candidatesCount, \_name, 0);

}

}

#3 Contract

pragma solidity ^0.4.2;

contract Election {

//Model candidate

struct Candidate {

uint id;

string name;

uint voteCount;

}

//Store accounts that voted

mapping(address => bool) public voters;

//Store candidates

mapping(uint => Candidate) public candidates; //Key Value pair

//Fetch candidates

//Store candidate count

uint public candidatesCount;

function Election() public {

addCandidate("Niharika");

addCandidate("Rachel");

addCandidate("Tom");

}

function addCandidate (string \_name) private {

candidatesCount++;

candidates[candidatesCount] = Candidate(candidatesCount, \_name, 0);

}

function vote (uint \_candidateId) public {

//Require that they haven't voted before

require(!voters[msg.sender]);

//Require a valid candidate

require(\_candidateId > 0 && \_candidateId <= candidatesCount);

//Record voter has voted

voters[msg.sender] = true;

//Update candidate vote count

candidates[\_candidateId].voteCount++;

}

}

#4

pragma solidity ^0.4.2;

contract Election {

//Model candidate

struct Candidate {

uint id;

string name;

uint voteCount;

}

//Store accounts that voted

mapping(address => bool) public voters;

//Store candidates

mapping(uint => Candidate) public candidates; //Key Value pair

//Fetch candidates

//Store candidate count

uint public candidatesCount;

//Voted event

event votedEvent (uint indexed \_candidateId);

function Election() public {

addCandidate("Niharika");

addCandidate("Rachel");

addCandidate("Tom");

}

function addCandidate (string \_name) private {

candidatesCount++;

candidates[candidatesCount] = Candidate(candidatesCount, \_name, 0);

}

function vote (uint \_candidateId) public {

//Require that they haven't voted before

require(!voters[msg.sender]);

//Require a valid candidate

require(\_candidateId > 0 && \_candidateId <= candidatesCount);

//Record voter has voted

voters[msg.sender] = true;

//Update candidate vote count

candidates[\_candidateId].voteCount++;

//Trigger voted event

votedEvent(\_candidateId);

}

}