Project Name: Project 1: Voting System	**Team#4**
Test Stage: Unit x **System** __	**Test Date: 3/23/2022**
Test Case ID#: Candidate_constructor_1	**Name(s) of Testers:** Charlie Nazarian, Haneesha Kella
Test Description: Testing the Candidate constructor with a parameter for the name	
Automated: yes___ **no** x	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** Tests are stored in CandidateTest.cpp. The function being tested is
Results: Pass x **Fail**_____	Candidate(std::string name) (constructor)
Preconditions for Test: Candidate.h is included. The input parameter is of type string	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create Candidate Object with a name	Candidate cand("John")	–	–	
2	Check to see that the name is correct	cand.getName()	"John"	"John"	
3	Check to see that the numVotes is correct	cand.getNumVotes()	0	0	
4					

Post condition(s) for Test:

A candidate object is created with the name variable being equal to the input parameter and numVotes = 0

Project Name: The project #, name of your system, and the team#

Test Stage: Indicate whether it is a unit test or a system test.

Test Date: The date the test was performed.

Test Case ID#: A unique ID is required. Decide on a naming convention and use numbering. Example: Ballot_Shuffle_1

Name(s) of Testers: List the names of anyone involved in running this test case.

Test Description: Describe briefly the test objective.

Automated: Indicate if the test is completely automated or being checked manually. (If you have methods running the tests and checking results, select "yes". If you are manually checking results, indicate manual by selecting the "no.")

Results: Indicate if the test passed or failed.

Step #: You will be listing the test steps in order. This number is the step number in the process.

Test Step Description: Details of the test step.

Test Data: What the test data will be for this step. Be clear on what the input data will be. If using a specific file, be clear on the name.

Expected Result: What result are you expecting from the program component or system.

Actual Result: What result were returned based on the test.

Post condition for Test: What will be true after the test has been run? Has the state of the system changed in any way?

Notes: Comments and notes for you and your team members.