

Project Name: Project 1: Voting System		Team# 4
Test Stage: Unit <u> x </u> System <u> </u>		Test Date: 3/24/22
Test Case ID#: Party_constructor_3		Name(s) of Testers: Charlie Nazarian, Haneesha Kella
Test Description: Testing the Party constructor with a single argument – a candidates vector		
Automated: yes <u> x </u> no <u> </u>		Indicate where are you storing the tests (what file) and the name of the method/functions being used. Tests are stored in PartyTest.cpp. Function being tested is Party(std::vector<Candidates> candidates) (constructor)
Results: Pass <u> x </u> Fail <u> </u>		
Preconditions for Test: Party.h and Candidate.h are included in the file, the input parameter is of type Candidate vector		

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create Party Object with Candidate vector input	Party Democrat(candidates)	–	–	Let the vector "candidates" contain Candidate objects "Obama" and "Biden"
2	Compare actual name with expected name	Democrat.getPartyName()	Expected name: "Independent"	Actual name: "Independent"	
3	Compare actual numVotes with expected numVotes	Democrat.getNumVotes()	Expected numVotes: 0	Actual numVotes: 0	
4	Compare candidate vector size with 0	Democrat.getCandidates().size()	Expected size: 2	Actual size: 2	
	Compare candidate names to expected candidate names	Democrat.getCandidates()[i]	Expected name: "Obama" (if i=0), "Biden" (if i=1)	Actual name: "Obama" (when i=0), "Biden" (when i=1)	

Post condition(s) for Test:

A Party object is created with the candidates vector being set to the candidates vector entered as the parameter, numVotes = 0, and a partyName of "Independent".

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.