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.

Project Name: Project 1: Election Voting System **Team# 3**

Test Stage: Unit _1_ System __

Test Date: 3/24/2023

Test Case ID#: testPopulateParties
Test Description: testing the function populateParties of the class CPL to see if parties are correctly populated with the correct information.

Name(s) of Testers: Chiemeka

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes___

Results: Pass____

Preconditions for Test:
The File called "file" must be provided and open properly. It must also contain the correct format of a CPL election file. The audit file must be properly made in the constructor

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	The user runs the test file, the set up is the file is open and a cpl object is created.	A cpl object created from the file. The cpl object is populated with data about the parties in its constructor	Number of parties = 7 Parties: Democratic Republican New Wave Reform Green Independent Test	Number of parties = 7 Parties: Democratic Republican New Wave Reform Green Independent Test	The formatting of expected does not matter, we are looking for if the answers(such as 7) are correct as the actual results. Parties is a string with the party names separated by a space
3					
4					

Post condition(s) for Test:
The cpl object now has objects of parties. The system has changed in that the class variable parties is no longer empty, numParties has a value.

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __ **Test Date: 3/24/2023**

Test Case ID#: testPopulateCandidate **Name(s) of Testers: Chiemeka**

Test Description: testing the function populateCandidates of the class CPL to see if candidates are correctly populated with the correct information.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes___

Results: Pass _____

Preconditions for Test:
The File called "file" must be provided and open properly. It must also contain the correct format of a CPL election file. The cpl object must already have party objects populated beforehand. The audit file must be properly made in the constructor

Step # 1	Test Step Description	Test Data	Expected Result	Actual Result	Notes
2	The user runs the test file, the set up is the file is open and a cpl object is created.	A cpl object created from the file. The cpl object is populated with data about the parties in its constructor. Then the constructor calls populateCandidates, and the candidates are populated with information	Foster Volz Pike Green Xu Wang Jacks Rosen McClure Berg Zheng Melvin Peters Tester	Foster Volz Pike Green Xu Wang Jacks Rosen McClure Berg Zheng Melvin Peters Tester	The expected candidates is a string with the candidates names separated by space for each party. It does not include the new line for each party, that is for readability
3					
4					

Post condition(s) for Test:

The cpl object has parties that are populated with candidates. The system has changed in that each party has an array of candidates.

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __ **Test Date: 3/24/2023**

Test Case ID#: testPopulateBallots
Test Description: testing the function populateBallots of the class CPL to see if ballots are correctly populated with the correct information.

Name(s) of Testers: Chiemeka

Automated: yes

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Results: Pass _____

Preconditions for Test:
The File called "file" must be provided and open properly. It must also contain the correct format of a CPL election file. The cpl object must already have party objects populated beforehand. The file must be opened at the point where the ballots are at. The audit file must be properly made in the constructor

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	The user runs the test file, the set up is the file is open and a cpl object is created.	A cpl object created from the file. The cpl object is populated with data about the parties and candidates in its constructor. Then the constructor calls populateBallots, and the ballot objects are populated with information	Ballot 0: 1,,,,,, Ballot 1: ,,,,,,1 Ballot 2: ,1,,,,, Ballot 3: ,,,,1,, Ballot 4: ,,,,,1, Ballot 5: ,,,1,,, Ballot 6: ,,1,,,, Ballot 7: 1,,,,,, Ballot 8: ,1,,,,,	Ballot 0: 1,,,,,, Ballot 1: ,,,,,,1 Ballot 2: ,1,,,,, Ballot 3: ,,,,1,, Ballot 4: ,,,,,1, Ballot 5: ,,,1,,, Ballot 6: ,,1,,,, Ballot 7: 1,,,,,, Ballot 8: ,1,,,,,	The newlines do not matter, what is being compared is the strings without the newline
3					
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __

Test Date: 3/24/2023

Test Case ID#: testgetNumParties
Test Description: testing the function getNumParties of the class CPL to see if the number of parties are returned correctly.

Name(s) of Testers: Chiemeka

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes

Results: Pass _____

Preconditions for Test:
The File called "file" must be provided and open properly. It must also contain the correct format of a CPL election file. The cpl object must already have party objects populated beforehand.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	The user runs the test file, the set up is the file is open and a cpl object is created.	A cpl object created from the file is populated with objects of parties, candidates, and ballots. The function returns the number of parties.	7	7	
3					
4					

Post condition(s) for Test:

System has not changed.

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __

Test Date: 3/24/2023

Test Case ID#: testgetParty

Name(s) of Testers: Chiemeka

Test Description: testing the function getParty of the class CPL to see if the array "parties" is returned correctly. Its memory address is returned, and this is used to test if the contents of the parties(names) is the same.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes___

Results: Pass ____

Preconditions for Test:
The File called "file" must be provided and open properly. It must also contain the correct format of a CPL election file. The cpl object must already have party objects populated beforehand.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	The user runs the test file, the set up is the file is open and a cpl object is created.	The cpl parties array	Democratic Republic New Wave Reform Green Independent Test	Democratic Republic New Wave Reform Green Independent Test	We are collecting the parties names in a string by storing the result of getParty in a Party array.
3					
4					

Post condition(s) for Test:

The system has not changed

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __ **Test Date: 3/24/2023**

Test Case ID#: testgetNumCandidates **Name(s) of Testers: Chiemeka**

Test Description: testing the function getNumCandidates of the class CPL to see if the number of candidates are returned correctly.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java

Automated: yes **Uses: testCPL.csv**

Results: Pass

Preconditions for Test:
The File called "file" must be provided and open properly. It must also contain the correct format of a CPL election file. The cpl object must already have party and candidate objects populated beforehand. The calculation of number of candidates is done during the populating of the candidates

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1		The number of candidates total, obtained by the function getNumCandidates and identified during populateCandidates	14	14	
2					
3					
4					

Post condition(s) for Test:

The system has not changed

Test Stage: Unit _1_ System __

Test Date: 3/24/2023

Test Case ID#: testgetTotalSeats

Name(s) of Testers: Chiemeka

TTest Description:

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes

Results: Pass _____

Preconditions for Test:
The File called "file" must be provided and open properly. It must also contain the correct format of a CPL election file. The cpl object must already have party and candidate objects populated beforehand. The calculation of number of candidates is done during the populating of the candidates

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	call getTotatSeats on cpl object, ,see if it returns what is expected	The cpl object	6	6	
3					
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __

Test Date: 3/24/2023

Test Case ID#: testgetTotalSeats

Name(s) of Testers: Chiemeka

Test Description:

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes

Results: Pass

Preconditions for Test:

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2					
3					
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __

Test Date: 3/25/2023

Test Case ID#: testgetBallots

Name(s) of Testers:

Test Description:

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes___

Results: Pass _____

Preconditions for Test:
CPL object must be created and populated with data from the election file, specifically the ballots.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2		getBallots returns an array of ballot objects	Ballot 0: 1,,,,,, Ballot 1: ,,,,,,1	Ballot 0: 1,,,,,, Ballot 1: ,,,,,,1	The newlines do not matter, what is being compared is the

			Ballot 2: ,1,,,,, Ballot 3: ,,,,1,, Ballot 4: ,,,,,1, Ballot 5: ,,,1,,, Ballot 6: ,,1,,,, Ballot 7: 1,,,,,, Ballot 8: ,1,,,,,	Ballot 2: ,1,,,,, Ballot 3: ,,,,1,, Ballot 4: ,,,,,1, Ballot 5: ,,,1,,, Ballot 6: ,,1,,,, Ballot 7: 1,,,,,, Ballot 8: ,1,,,,,	strings without the newline
3					
4					

Post condition(s) for Test:

The system has not changed

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __ **Test Date: 3/24/2023**

Test Case ID#: testAssignBallots **Name(s) of Testers: Chiemeka**

Test Description:

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes___

Results: Fail_____

Preconditions for Test:
CPL object must be created and populated with data from the election file, specifically the audit file, the array of parties objects, the array of ballot objects.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2		The audit file will be written to about the results of the assignments. The class variable array of ballots. The class variable array of parties			This test was changed last minute, it may not work as expected. But there was not enough time to test it. Thus, it is labeled a failed test
3					
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __ **Test Date: 3/24/2023**

Test Case ID#: testdistributeSeatsCoinToss **Name(s) of Testers: Chiemeka**

Test Description:
Tests if the function coin toss in the distributeSeats function is fair

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes

Results: Pass

Preconditions for Test:
CPL object must be created and populated with data from the election file, specifically the audit file, the array of parties objects, the array of ballot objects. Ballots must be assigned by calling assignBallots(), this occurs in the run() function. The election file must be in the proper. There must be a tie between two parties that result in calling the coin toss

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1		cpl array of parties, number of ballots, total seats, and ballots array of each party, two parties with the same number of votes remaining and there is only 1 seat to give.	"fair around 50-50"	"fair around 50-50"	
2					
3					
4					

Post condition(s) for Test:

The system has not changed

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __ **Test Date: 3/25/2023**

Test Case ID#: testDuplicates **Name(s) of Testers: Chiemeka**

Test Description:
Tests if the correct number of duplicates in votes remaining is found from a starting index, in a row.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes

Results: Pass

Preconditions for Test:
CPL object must be created and populated with data from the election file, specifically the audit file, the array of parties objects, the array of ballot objects. Ballots must be assigned by calling assignBallots(), this occurs in the run() function. The election file must be in the proper. There must be a tie between 3 or more parties, as duplicates() will be called to assist in the pooling function

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2		cpl array of parties, number of ballots, total seats, remaining seats, and ballots array of each party. 3 or more parties have the same amount of votes remaining and there are less seats to give.	Expected at index 0-1: 2 Expected at index 2-6: 5	Expected at index 0-1: 2 Expected at index 2-6: 5	The formatting is not what matters, it is the values 2 and 5. This means that from index 0, there are 2 duplicates(same number of remaining votes). And at index 2, there are 5 duplicates in a row.
3					
4					

Post condition(s) for Test:

The system has not changed

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __ **Test Date: 3/25/2023**

Test Case ID#: testCalculateRemainingVotes **Name(s) of Testers: Chiemeka**

Test Description:
Tests if the remaining votes for each party are correctly calculated after the first round

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes

Results: Pass

Preconditions for Test:
CPL object must be created and populated with data from the election file, specifically the audit file, the array of parties objects, the array of ballot objects. Ballots must be assigned by calling assignBallots(), this occurs in the run() function. The election file must be in the proper. The election/program must be at the stage of after the first round and between the second round.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes

1					
2		cpl array of parties, number of ballots, total seats, remaining seats, and ballots array of each party. Each party has been give some amount of seats, they have their initial votes	0 0 1 1 1 1 1	0 0 1 1 1 1 1	
3					
4					

Post condition(s) for Test:

The system has changed in that the remaining votes for each party has been computed and assigned. This is the number of votes that will be used in the second round.

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __

Test Date: 3/26/2023

Test Case ID#: testPartyGetName

Name(s) of Testers: Chiemeka

Test Description:
Tests if the correct name of the party is returned

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes___

Results: Pass

Preconditions for Test:
A party object is created and provided a name

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2		A party object is created and given a name	"Obama"	"Obama"	
3					
4					

Post condition(s) for Test:

The system has not changed

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __

Test Date: 3/26/2023

Test Case ID#: testAddSeatParty

Name(s) of Testers: Chiemeka

Test Description:
Tests if a given amount is added to a party's seat count correctly

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Results: Pass

Preconditions for Test:
A party object is created

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2		A party object is created and given a name	1	1	The 1 indicates that a seat has been added to the party
3					
4					

Post condition(s) for Test:

The system has changed in that the party's total seats has been increased by the value provided to the user, in this case, 1.

Project Name: Project 1: Voting System **Team# 3**

Test Stage: Unit _1_ System __ **Test Date: 3/26/2023**

Test Case ID#: tesetremainderVotes **Name(s) of Testers: Chiemeka**

Test Description:
Tests setting the party's remaining votes to a new value

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes

Results: Pass

Preconditions for Test:
A party object is created, its remaining votes count is changed

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2		A party object is created, its remaining votes count changes.	remainder votes: 1	remainder votes: 1	
3					
4					

Post condition(s) for Test:

The system has changed in that the party's remaining votes have been changed.

Test Stage: Unit _1_ System __

Test Date: 3/26/2023

Test Case ID#: testAddSeatCandidate

Name(s) of Testers: Chiemeka

Test Description:
Tests adding a seat to a candidate correctly

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes___

Results: Pass

Preconditions for Test:
A candidate object is created

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2		A candidate object is created, Bob of the Democrats	1	1	The 1 represents the seats of the candidates. The actual result is obtained via getSeats()
3					
4					

Post condition(s) for Test:

The system has changed in that the candidates total number of seats has increased by 1

Test Stage: Unit _1_ System __

Test Date: 3/26/2023

Test Case ID#: testgetCandidateName

Name(s) of Testers: Chiemeka

Test Description:
Tests getting a candidate's name through the getCandidateName function

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test methods in ElectionTestCPL.java
Uses: testCPL.csv

Automated: yes

Results: Pass

Preconditions for Test:
A candidate object is created

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2		A candidate object is created, Bob of the Democrats	"Bob"	"Bob"	
3					
4					

Post condition(s) for Test:

The system has not changed