The PBI, the Task Description (from Sprint Log) with Unique Testing Number	PBI #1 - Bug: File Handler does not handle non-existent files cleanly Task Description: Update Main function to handle nonexistent files cleanly. Testing Number: #001
Team Member(s) Responsible	Robert, Alex
Inputs	Files which do not exist in the src directory, such as “nonexistent.csv”.
Tests	Test whether the program will continue and attempt to process this file or whether the program will request a new file to be entered (as desired). Continue to enter nonexistent files and observe behavior.
Outputs	User is prompted in the terminal to enter a new filename until a valid filename is entered.
Passed or Failed	Passed
Date	4/30/23

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	PBI #2 - Bug: Out of bounds error in addToResults in CPL Task Description: Need to fix the addToResults function that would sometimes raise an out of bounds error for files where there are many ties Testing Number: #002
Team Member(s) Responsible	Robert, Alex
Inputs	“testCPLAllTie.csv, testCPL.csv, testCPLTwoTie.csv”
Tests	- runNormalElectionTest(), runAllTieElectionTest(): For this test, we would simply need to check if the program raised any errors during execution, since originally the bug terminated the program with errors.
Outputs	All tests ran without errors
Passed or Failed	Passed
Date	4/30/23

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	PBI #3 - Bug: Break a Tie Task Description: Need to fix the break a tie functionality in CPL which would not account for all candidates Testing Number: #003
Team Member(s) Responsible	Robert, Alex
Inputs	“testCPLAllTie.csv, testCPLTwoTie.csv”
Tests	- runAllTieElectionTest(): Since tied files were the only ones causing issues, a test was written to run CPL election on these files 100s of times, and during each iteration the program would keep track of how many times each candidate won. At the very end, the total number of wins for each candidate should be around the same range.
Outputs	All parties and their candidates won around the same number of times.
Passed or Failed	Passed
Date	4/30/23

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	PBI #4 - Input Multiple Files (IR, CPL) Task Description: Test to make sure that multiple files are being read in and processed correctly. Testing Number: #004
Team Member(s) Responsible	Robert, Alex
Inputs	“testIR.csv, testIR_file1.csv…file3.csv, testIRAllTied.csv, testIRTied_file1.csv…file4.csv”
Tests	- IR runElection(): This test checked to see if running a 1 file IR election and a multi-file IR election with the original file split up would have the same results. At the same time, tests were performed to see if tie breaking would work properly on multiple files.
Outputs	testIR.csv and file 1-3 of the multi-file IR run returned the same winning candidate. testIRAllTied.csv and file1-4 of the multi-file IR returned the same ~equal ratio wins for each of the candidates.
Passed or Failed	Passed
Date	4/30/23

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	PBI #5 - Election Table (IR) Task Description: Update printElectionResults() to match the desired format. Testing Number: #005
Team Member(s) Responsible	Gideon
Inputs	testIR.csv
Tests	Check that runElection() prints the desire format to the screen
Outputs	Prints election results in desired format
Passed or Failed	Passed
Date	4/30/23

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	PBI #6 - Process a file (PO) Task Description: Test to see if PO candidate information is being stored correctly and is compared in the right way. Testing Number: #006
Team Member(s) Responsible	Robert, Alex
Inputs	`POCandidate candidate = new POCandidate("name", 0);` `POCandidate other = new POCandidate("other", 1);`
Tests	- getWinnerTest() to see if the set winner function of PO candidate works properly. - comparePOTest(): test to see is the compareTo function in PO candidate works to properly compare two candidate's number of votes and winner status - POCandidateConstructoreTest(): test to make sure that the constructor is actually instantiating an object for this candidate and not returning null. - processGoodFileTest(), processFileWithNoBallotsTest(), and processManyFilesTest() all test that the information of a PO election is read in correctly. This tests whether the correct candidates are stored, the correct number of candidates are stored, and the correct number of ballots are stored.
Outputs	All the tests outputted what was expected of a correct program, which means the correct values were returned when comparing two candidates with different and equal votes, and the constructor does not create null objects. The tests passed, correctly processing the files for the correct number of ballots, candidates, etc.
Passed or Failed	Passed
Date	4/30/23

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	PBI #7 - Determine the winner of a PO election Task Description: - Add functionality to determine the correct winner from the stored ballots Testing Number: #007
Team Member(s) Responsible	Gideon
Inputs	- A file with data about the PO election in question
Tests	1. Test that a PO election with standard data(all candidates received different amounts of votes) matches the expected results 2. Test that a PO election where two candidates were tied matches the expected results 3. Test that a PO election where there was only one candidate matches the expected results 4. Test that a PO election where there were no ballots matches the expected results
Outputs	OK or error message if result and expected result do not match
Passed or Failed	Passed
Date	4/30/23