

<p><b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b></p> <p><b>PBI 001:</b> Bring in multiple files into the election system  <b>Task Description:</b> Modify the Election.jav to bring in multiple files for CPL and IR election.  <b>Testing Number:</b> <b>W1_01</b></p>
<p><b>Team Member(s) Responsible:</b></p> <p>Chiemeka Nwakama</p>
<p><b>Inputs:</b></p> <ol style="list-style-type: none"> <li>1. Test files for CPL and IR election stored in the /testing folder</li> <li>2. Multiple files input into the terminal for either CPL or IR election</li> </ol>
<p><b>Tests:</b></p> <ol style="list-style-type: none"> <li>1. The selected election type is running without errors</li> <li>2. The election results is display to the terminal</li> <li>3. Audit file is generated</li> </ol>
<p><b>Outputs:</b></p> <ol style="list-style-type: none"> <li>1. Multiples files were able to process without errors OR Error message display for invalid files name</li> <li>2. Audit.txt was generated for the selected election</li> <li>3. The correct results was displayed</li> </ol>
<p><b>Passed or Failed</b></p>
<p><b>April 28, 2023</b></p>

<p><b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b></p> <p><b>PBI 002:</b> Bring in multiple files into the IR election system  <b>Task Description:</b> Modify the IR.java to bring in multiple files for IR election.  <b>Testing Number:</b> <b>W1_02</b></p>
<p><b>Team Member(s) Responsible:</b></p> <p>Phuong Lieu</p>
<p><b>Inputs:</b></p> <ol style="list-style-type: none"> <li>1. An IR object containing a single file (IRfile1.csv)</li> <li>2. An IR object containing multiple (IRfile1.csv, IRfile2.csv, IRfile3.csv)</li> </ol>
<p><b>Tests:</b></p> <p>* Please see the Previous IR_Testing.docx for details on these JUnit tests.</p> <ul style="list-style-type: none"> <li>• testInitialData - test the initial data is used in the constructor to see if it populated the</li> </ul>

<p>data correctly</p> <ul style="list-style-type: none"> <li>• testpopulateData - test to see if data is populated correctly for the ballots</li> <li>• testgetCandidates - test that candidates are correctly populated</li> <li>• testgetNumCandidates - test that the total number of candidates are correctly populated</li> <li>• testgetNumBallots - test that the total ballots are correctly stored</li> <li>• testUpdateCurrentVote - test that the candidate's vote are correctly updated</li> <li>• testsSetElectionStatus - test that the leading candidate is correctly updated</li> <li>• testMakeLoser - test that the losers in the election are updated correctly</li> <li>• testCoinToss - test that the candidate choosing in a tie election is fair</li> <li>• testPoolSelect - test that the candidate choosing in a tie election is fair</li> </ul>
<p><b>Outputs:</b></p> <ol style="list-style-type: none"> <li>1. JUnit tests passed</li> <li>2. Multiples files were able to process without errors OR Error message display for invalid files name</li> <li>3. Audit.txt was generated for the selected election</li> <li>4. The correct election results is displayed</li> </ol>
<p><b>Passed or Failed</b></p>
<p><b>April 28, 2023</b></p>

<p><b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b></p> <p><b>PBI 003:</b> Process multiple files into the CPL election system  <b>Task Description:</b> Process multiple files into the CPL election system  <b>Testing Number:</b> W1_03</p>
<p><b>Team Member(s) Responsible:</b></p> <p>Sidney Nguyen</p>
<p><b>Inputs:</b></p> <ol style="list-style-type: none"> <li>1. A CPL object containing a single file</li> <li>2. A CPL object containing three files (testCPL.csv, CPLfile1.csv, CPLfile2.csv)</li> </ol> <p>The CPL objects were set up before the tests, and all the tests used specific aspects of the CPL objects.</p>
<p><b>Tests:</b></p> <ul style="list-style-type: none"> <li>• JUnit tests: <ul style="list-style-type: none"> <li>○ testPopulateParties - tests that parties of multiple files are correctly populated.</li> <li>○ testPopulateParties - tests that parties of a single file are correctly populated even with the multiple file feature.</li> <li>○ testPopulateCandidates - tests that candidates of multiple files are correctly populated</li> </ul> </li> </ul>

- testPopulateCandidates - tests that Candidates of a single file are correctly populated even with the multiple file feature.
- testPopulateBallots - tests that all ballots from multiple files are combined together in one data structure correctly
- testPopulateBallots - tests that ballots from a single file are correctly populated in one data structure correctly even with the multiple file feature
- testgetNumParties - tests that the number of parties remains the same even with multiple files processed
- testgetNumParties - tests that the number of parties is correct with a single file even with the multiple file feature
- testgetParty - tests the actual parties are correct even with multiple files processed
- testgetParty - tests the actual parties are correct for a single file even with the multiple file feature
- testgetNumCandidates - tests the number of candidates remains the same even with multiple files processed
- testgetNumCandidates - tests the number of candidates is correct with a single file even with the multiple file feature
- testgetTotalSeats - tests the total number of seats remains the same even with multiple files processed
- testgetTotalSeats - tests the total number of seats is correct with a single file even with the multiple file feature
- testAssignBallots - tests the assignment of ballots to parties is correct and is a secondary test to populating the ballots correctly from multiple files
- testAssignBallots - tests the assignment of ballots is correct from a single file even with the multiple file feature

**Outputs:**

1. JUnit tests passed
2. File(s) were able to process without error
  - a. The correct results are displayed
  - b. An audit file is generated with correct information
3. A file contained an error in one or more aspect and an error message is displayed
  - a. The incorrect results may be displayed
  - b. An audit file is generated with incorrect information

**Passed or Failed**

**April 26th, 2023**

**The PBI, the Task Description (from Sprint Log) with Unique Testing Number:**

**PBI 004:** A table that shows each round of the IRV

**Task Description:** Modify the IR.java to add new functions to output a table display for the IR election process.

**Testing Number:** W1\_04

<b>Team Member(s) Responsible:</b>  Ted Casey
<b>Inputs:</b> 1. An IR object containing a single file (IRfile1.csv)
<b>Tests:</b> <ul style="list-style-type: none"> <li>JUnit test:             <ul style="list-style-type: none"> <li>testTable - test to see if the table were able to produce with correct values</li> </ul> </li> <li>System test:             <ul style="list-style-type: none"> <li>Check if the display is able to produced to the terminal</li> <li>Check if the information in the election is correct</li> <li>Check if election process (votes transfer) is correct</li> </ul> </li> </ul>
<b>Outputs:</b> <ol style="list-style-type: none"> <li>JUnit test passed</li> <li>Multiples files were able to process without errors OR Error message display for invalid files name</li> <li>Audit.txt was generated for the selected election</li> <li>The correct election results is displayed</li> <li>The table is displayed to the terminal</li> </ol>
<b>Passed or Failed</b>
<b>April 28, 2023</b>

### Incomplete

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>  <b>PBI 005:</b> View what votes are added and what votes are decreased <b>Task Description:</b> Modify the IR.java to track each round's vote total for each candidates in IR election process. <b>Testing Number:</b> W1_05
<b>Team Member(s) Responsible:</b>  Ted Casey
<b>Inputs:</b> 1. An IR object containing a single file (IRfile1.csv)
<b>Tests:</b> <ul style="list-style-type: none"> <li>JUnit test:             <ul style="list-style-type: none"> <li>testTable - test to see if the table were able to produce with correct values</li> </ul> </li> <li>System test:             <ul style="list-style-type: none"> <li>Check if the display is able to produced to the terminal</li> </ul> </li> </ul>

<ul style="list-style-type: none"> <li>○ Check if the information in the election is correct</li> <li>○ Check the process of the votes being added and decreased over each round</li> <li>○ Check if election process (votes transfer) is correct</li> </ul>
<b>Outputs:</b> <ol style="list-style-type: none"> <li>1. JUnit test passed</li> <li>2. Multiples files were able to process without errors OR Error message display for invalid files name</li> <li>3. Audit.txt was generated for the selected election</li> <li>4. The correct election results is displayed</li> <li>5. The table is displayed to the terminal</li> </ol>
<b>Passed or Failed</b>
<b>April 28, 2023</b>

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>  <b>PBI 006:</b> Process multiple files into the PO election system <b>Task Description:</b> Create PO.java to process multiple files in construction. <b>Testing Number:</b> W2_06
<b>Team Member(s) Responsible:</b>  Chiemeka Nwakama
<b>Inputs:</b> <ol style="list-style-type: none"> <li>1. An PO object containing single files (POfile1.csv)</li> <li>2. An PO object containing multiple files (POfile1.csv, POfile2.csv, POfile3.csv)</li> </ol>
<b>Tests:</b> <ul style="list-style-type: none"> <li>• JUnit test: testTable <ul style="list-style-type: none"> <li>○ test to see if the table were able to produce with correct values</li> <li>○ Check if election can bring in a single file</li> <li>○ Check if election can bring in multiple files</li> <li>○ Check file names corrected processed</li> </ul> </li> <li>• System test: <ul style="list-style-type: none"> <li>○ Check if election can process a single file</li> <li>○ Check if election can process multiple files</li> </ul> </li> </ul>
<b>Outputs:</b> <ol style="list-style-type: none"> <li>1. JUnit test passed</li> <li>2. File(s) were able to process without error <ol style="list-style-type: none"> <li>a. The correct results are displayed</li> <li>b. An audit file is generated with correct information</li> </ol> </li> <li>3. A file contained an error in one or more aspect and an error message is displayed</li> </ol>

<ul style="list-style-type: none"> <li>a. The incorrect results may be displayed</li> <li>b. An audit file is generated with incorrect information</li> </ul>
<b>Passed or Failed</b>
<b>April 28, 2023</b>

#### Incomplete

<p><b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b></p> <p><b>PBI 007:</b> Candidate votes are accounted for</p> <p><b>Task Description:</b> Modify the IR.java with new functions to account invalidate vote that is not ranked for at least half the candidates.</p> <p><b>Testing Number:</b> <b>W2_07</b></p>
<p><b>Team Member(s) Responsible:</b></p> <p>Ted Casey</p>
<p><b>Inputs:</b></p> <ol style="list-style-type: none"> <li>1. An IR object containing a single file (IRfile1.csv)</li> </ol>
<p><b>Tests:</b></p> <ul style="list-style-type: none"> <li>• System test: <ul style="list-style-type: none"> <li>○ Check if the display is able to produced to the terminal</li> <li>○ Check if the information in the election is correct</li> <li>○ Check the process of the votes being added and decreased over each round</li> <li>○ Check if election process (votes transfer) is correct</li> <li>○ Check if the invalidate votes stored correctly in the</li> </ul> </li> </ul>
<p><b>Outputs:</b></p> <ol style="list-style-type: none"> <li>1. Multiples files were able to process without errors OR Error message display for invalid files name</li> <li>2. Audit.txt was generated for the selected election</li> <li>3. The correct election results is displayed</li> <li>4. The table is displayed to the terminal</li> </ol>
<b>Passed or Failed</b>
<b>April 28, 2023</b>

#### Incomplete

<p><b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b></p> <p><b>PBI 008:</b> IRV ballots to have at least half the candidates ranked for it to be valid</p> <p><b>Task Description:</b> Modify the IR.java to change populateData() to exhaust invalid ballots and add those to a file.</p>
---

<b>Testing Number:</b> W2_08
<b>Team Member(s) Responsible:</b>  Ted Casey
<b>Inputs:</b> 1. An IR object containing multiple files (IRfile1.csv)
<b>Tests:</b> <ul style="list-style-type: none"> <li>JUnit test: testExhaustIncomplete() <ul style="list-style-type: none"> <li>Check if exhausted votes are extracted out of ballots</li> <li>Check if the candidates votes are updated correctly</li> </ul> </li> </ul>
<b>Outputs:</b> 1. Junit test passed
<b>Passed or Failed</b>
<b>April 28, 2023</b>

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>  <b>PBI 009:</b> A new file containing all of the invalidated votes <b>Task Description:</b> Produce the invalidated_dateofelection.xxx containing invalidated votes from populateData(). <b>Testing Number:</b> W2_09
<b>Team Member(s) Responsible:</b>  Phuong Lieu
<b>Inputs:</b> 1. An IR object containing multiple files (IRfile1.csv, IRfile2.csv, IRfile3.csv)
<b>Tests:</b> <ul style="list-style-type: none"> <li>System test: <ul style="list-style-type: none"> <li>Check if the invalidated_ddateofelection.txt is produced</li> <li>Check if the file contain the correct information on invalid votes</li> </ul> </li> </ul>
<b>Outputs:</b> 1. File(s) were able to produced without error <ul style="list-style-type: none"> <li>The correct results are displayed</li> <li>An audit file is generated with correct information</li> </ul>
<b>Passed or Failed</b>
<b>April 28, 2023</b>

<p><b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b></p> <p><b>PBI 010:</b> The PO election determines a single winner when there are no ties</p> <p><b>Task Description:</b> Modify the PO.java with new functions that allows to to declare candidate winner when there is no tie.</p> <p><b>Testing Number:</b> W2_10</p>
<p><b>Team Member(s) Responsible:</b></p> <p>Chiemeka Nwakama</p>
<p><b>Inputs:</b></p> <ol style="list-style-type: none"> <li>1. An PO object containing single files (POfile1.csv)</li> <li>2. An PO object containing multiple files (POfile1.csv, POfile2.csv, POfile3.csv)</li> </ol>
<p><b>Tests:</b></p> <ul style="list-style-type: none"> <li>• Junit tests: <ul style="list-style-type: none"> <li>○ selectWinner - test that the corect winner is selected in the election for a single file</li> <li>○ selectWinner - test that the corect winner is selected in the election for multiple files</li> </ul> </li> </ul>
<p><b>Outputs:</b></p> <ol style="list-style-type: none"> <li>1. Junit tests passed</li> </ol>
<p><b>Passed or Failed</b></p>
<p><b>April 28, 2023</b></p>

<p><b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b></p> <p><b>PBI 011:</b> An unbiased system to break ties for PO election</p> <p><b>Task Description:</b> Modify the PO.java with new functions, coinToss() or poolSelect(), that allows unbiased system to to declare candidate winner when there is a tie.</p> <p><b>Testing Number:</b> W2_11</p>
<p><b>Team Member(s) Responsible:</b></p> <p>Chiemeka Nwakama</p>
<p><b>Inputs:</b></p> <ol style="list-style-type: none"> <li>1. An PO object containing single files (POfile1.csv)</li> <li>2. An PO object containing multiple files (POfile1.csv, POfile2.csv, POfile3.csv)</li> </ol>
<p><b>Tests:</b></p> <ul style="list-style-type: none"> <li>• JUnit tests: <ul style="list-style-type: none"> <li>○ testselectWinnerPool - test the fairness of pool select for candidates in a tie</li> </ul> </li> </ul>



<ul style="list-style-type: none"> <li>election with a single file</li> <li>○ testselectWinnerPool - test the fairness of pool select for candidates in a tie election with multiple files</li> <li>○ testselectWinnerCoinToss - test the fairness of coin toss for candidates in a tie election with a single file</li> <li>○ testselectWinnerCoinToss - test the fairness of coin toss for candidates in a tie election with multiple files</li> </ul>
<b>Outputs:</b> <ol style="list-style-type: none"> <li>JUnit tests passed</li> </ol>
<b>Passed or Failed</b>
<b>April 28, 2023</b>

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>  <b>PBI 012:</b> The PO election winner and stats displayed on the screen <b>Task Description:</b> Modify the PO.java with new functions, display(), that allows the results of the PO election to display on the terminal. <b>Testing Number:</b> W2_12
<b>Team Member(s) Responsible:</b>  Phuong Lieu
<b>Inputs:</b> <ol style="list-style-type: none"> <li>An PO object containing single files (POfile1.csv)</li> <li>An PO object containing multiple files (POfile1.csv, POfile2.csv, POfile3.csv)</li> </ol>
<b>Tests:</b> <ul style="list-style-type: none"> <li>System test: <ul style="list-style-type: none"> <li>○ Check if the election result is displayed to the terminal</li> <li>○ Check if correct election result is showing</li> <li>○ Check if the vote's percentages for each candidate is showing</li> </ul> </li> </ul>
<b>Outputs:</b> <ol style="list-style-type: none"> <li>Election result is displayed in the terminal</li> <li>Correct election result is showing</li> <li>Vote's percentages is showing for each candidate</li> </ol>
<b>Passed or Failed</b>
<b>April 28, 2023</b>

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>
--

**PBI 013:** Run the entire PO election

**Task Description:** Running and checking that PO election is running correctly.

**Testing Number:** W2\_13

**Team Member(s) Responsible:**

Chiemeka Nwakama

**Inputs:**

1. An PO object containing single files (POfile1.csv)
2. An PO object containing multiple files (POfile1.csv, POfile2.csv, POfile3.csv)

**Tests:**

- JUnit tests:
  - testPopulateCandidates - tests that candidates of multiple files are correctly populated
  - testPopulateCandidates - tests that Candidates of a single file are correctly populated even with the multiple file feature.
  - testPopulateBallots - tests that all ballots from multiple files are combined together in one data structure correctly
  - testPopulateBallots - tests that ballots from a single file are correctly populated in one data structure correctly even with the multiple file feature
  - testgetNumCandidates - tests the number of candidates remains the same even with multiple files processed
  - testgetNumCandidates 0 tests the number of candidates is correct with a single file even with the multiple file feature
  - testgetBallots - test that the number of parties is being stored correctly in a single file
  - testgetBallots - test that the number of parties is being stored correctly in multiple files
  - testAssignBallots - tests the assignment of ballots to parties is correct and is a secondary test to populating the ballots correctly from multiple files
  - testAssignBallots - tests the assignment of ballots is correct from a single file even with the multiple file feature
  - sortCandidates - test to see if ballots are assigned correctly to candidates via how many votes they have in a single file
  - sortCandidates - test to see if ballots are assigned correctly to candidates via how many votes they have in multiple files
  - testDuplicates - test to see if duplicates ballots are correctly found in a row for a single file
  - testDuplicates - test to see if duplicates ballots are correctly found in a row for multiple files
  - testGetNameandParty - test that the candidate's name and party are correctly stored for a single file
  - testGetNameandParty - test that the candidate's name and party are correctly stored for multiple files
  - testselectWinnerPool - test the fairness of pool select for candidates in a tie election with a single file
  - testselectWinnerPool - test the fairness of pool select for candidates in a tie

<ul style="list-style-type: none"> <li>election with multiple files</li> <li>○ testselectWinnerCoinToss - test the fairness of coin toss for candidates in a tie election with a single file</li> <li>○ testselectWinnerCoinToss - test the fairness of coin toss for candidates in a tie election with multiple files</li> <li>○ testAddVote - test the addVote() and getter for candidate's new vote for a single file</li> <li>○ testAddVote - test the addVote() and getter for candidate's new vote for multiple files</li> <li>○ selectWinner - test that the correct winner is selected in the election for a single file</li> <li>○ selectWinner - test that the correct winner is selected in the election for multiple files</li> </ul>
<b>Outputs:</b> 1. JUnit tests passed
<b>Passed or Failed</b>
<b>April 28, 2023</b>

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>  <b>PBI 013:</b> A GUI to select files <b>Task Description:</b> Implement the GUI window to select files for processing <b>Testing Number:</b> <b>W2_13</b>
<b>Team Member(s) Responsible:</b>  Sidney Nguyen
<b>Inputs:</b> 1. An PO object containing single files (POfile1.csv) 2. An PO object containing multiple files (POfile1.csv, POfile2.csv, POfile3.csv)
<b>Tests:</b> <ul style="list-style-type: none"> <li>• System test: <ul style="list-style-type: none"> <li>○ Check if the GUI window is showing</li> <li>○ Check if the user can select the file(s)</li> </ul> </li> </ul>
<b>Outputs:</b> 1. A pop-up (GUI) window is displayed to select files for processing
<b>Passed or Failed</b>
<b>April 28, 2023</b>