

After implementing my Ballot and Candidate classes, I created JUnit tests for each class.

For the ballot class, I tested the `getOfficeName()`, `addCandidate(Candidate c)`, `getCandidates()`, and `toString()` functions by creating a `BallotTest.java` class. I tested the `determineWinner()` function by running the program and intentionally letting one candidate win and compared the file output to when a candidate did not win.

The difference is shown below:

RESULTS - President		RESULTS - President	
-----		-----	
Donald J. Trump - Republican Party	0	Donald J. Trump - Republican Party	1
Jo Jorgensen - Libertarian	0	Jo Jorgensen - Libertarian	1
Joe Biden - Democrat	0	Joe Biden - Democrat	1
Bernie Sanders - Independent	1	Bernie Sanders - Independent	1
		NO WINNER	
WINNER: Bernie Sanders - Independent			

For the candidate class, I tested the `getName()`, `getAffiliation()`, `getVoteCount()`, `tallyVote()`, and `toString()` functions by creating a `CandidateTest.java` class.

Additionally, by running the program before implementing the `BallotReader` and `ResultWriter` classes, I verified that the program works as expected.

For the `BallotReader` class, I verified that the class worked as expected by creating an input file and inserting it into the program, and the choices dropdown looked and worked the same as the unimplemented `BallotReader` class that only had a list of candidates.

For the `ResultsWriter` class, I verified that the class worked as expected by outputting the results to files, and ensuring that the format matched as expected. I outputted 10 test files and adjusted the formatting as necessary. Each test file can be found in the "BallotWriter Output Tests" folder within the "Documents" folder. For the last test, I changed the number and names of the candidates, and office name, to ensure that the program worked as expected, regardless of the input as long as it is correctly formatted.