#	Description of Bug	Location of Bug	Steps to recreate bug or test case	Root Cause Analysis/Notes
1	checkWinner2() - doesn't pass and instead goes into an infinite loop	IRElectionTesting.java	(ensure all other files are compiling correctly before) 1. Run test checkWinner2()	Root cause: the way the election is set up has two simultaneous ties to break but neither are hitting and returning from the tiebreaker function correctly
2	checkRemoveBallot()	CandidateTesting.java	(ensure all other files are compiling correctly before) 1. Run test checkRemoveBallot()	Says getting element from ballot list is out of bounds. Likely because ballot hasn't been assigned to candidate correctly. Additional Note for dev team: add check in removeBallot() to ensure that not trying to remove from an empty list
3	checkGetInvalidatedBallots()	IRElectionTesting.java	(ensure all other files are compiling correctly before) 1. Run test checkGetInvalidatedBallots()	Ballots not being printed correctly, root cause is likely with remove ballots function
4	Incorrect handling of empty ballot preferences	translateBallot method in Ballot.java (tested in BallotTest.java)	1. Call translateBallot with the input string "1,,3," and a list of candidates. 2. Observe that the returned ArrayList does not have null for the missing preference (second	The method is expected to insert null for any missing preference to indicate no selection. The actual implementation incorrectly

			position).	assigns a non-null Candidate object instead of null. This likely results from improper parsing or mapping of empty strings to null values in the preference array.
5	Incorrect ArrayList size when translating an empty ballot	translateBallot method in Ballot.java (tested in BallotTest.java)	1. Call translateBallot with the input string ",,,," and a list of candidates. 2. Observe that the returned ArrayList has a size greater than the number of candidates, which should not be possible.	The method should return an ArrayList with a size equal to the number of candidates, all set to null. The actual implementation returns a list of size 5, indicating an off-by-one error or a misunderstanding in how empty rankings should be handled. This could be due to adding extra elements when encountering consecutive commas without associated rankings.
6	When you type a wrong input in the command line interface for the program, it doesn't prompt you for a new command and just closes instead	Structure of main in Controller.java	1. Run the program 2. Enter an invalid input or filename	Lack of a while loop in the main function that loops until the input is valid
7	OPL Election does not account for if candidates have the same number	OPLElection.java → displayResults() method	1. Run multiple iterations of the same OPL	Lack of tiebreaker checking for

	of votes in a party		election file where for a party, 2 candidates have the same number of votes 2. There will be a skew towards 1 candidate in getting the preferred seat if the party only has 1 seat assigned	Candidate List
8	IRElection does not create Audit files correctly	IRElection.java and FileProcessor.java → writeToAuditIR()	1. Run an IR election file through the controller 2. You will notice a corresponding audit file is NOT generated in the auditFile folder	Not getting the auditFileName correctly from class variable. Think it's due to a change in createAuditFile()