

Bug Description	Location	Steps to recreate bug / Test case #	Root cause analysis notes
Candidate and Party names are not parsed properly when they contain a first and last name or non-ASCII characters	Election.setCandidates()	<p>A Candidate/Party name contains characters other than ASCII letters.</p> <p>Shown in the testCandidatesWithLastNames() test case</p> <p>Also shown in TestSetCandidatesIRFullNames(), TestSetCandidatesIRPartyFullNames(), TestSetCandidatesOPLFullNames(), TestSetCandidatesOPLPartyFullNames(), TestSetCandidatesIRNonAscii(), and TestSetCandidatesOPLNonAscii test cases.</p>	<p>- Candidate and Party names are assumed to be delimited by characters other than ASCII letters.</p> <p>- If stripping the whitespace from all names causes them to only contain ASCII letters, the program functions as intended, albeit with the names concatenated (i.e. Bob Par becomes BobPar).</p> <p>Not sure if candidates would ever contain more than one name or non-ASCII characters, but just wanted to make you aware that we are aware of this issue.</p>
If there are less candidates than seats to be filled, the program runs in an infinite loop	OpenPartyListingElection.determineWinner()	Less candidates than seats to be filled	-We assumed that there wouldn't be an election with less candidates than seats to be filled, so we used a while loop when distributing based on remainders. This is the cause of this error, though we don't think it will be a problem because who would run an election with more seats to be filled than candidates on the ballot?

If a party is awarded more seats than they have candidates in their party, these seats are distributed in a round robin fashion based on the remainder to other candidates.	OpenPartyListingElection.determineWinner()	OPLSpecialTest in the OPLTest.java file	-This may not necessarily even be a bug, but we were just unsure as to how exactly to handle this scenario, so we had our algorithm run the same as it would, just giving out the seats on a round robin scheduler for the remainders.
Taking in files requires the filePath rather than just the file name	ElectionDriver	When passing in the file into the command line	The way we had our directory set up, we had to give it a file path since it uses a scanner object. If it is being run from the same directory this may not be an issue, but we wanted to make you aware.
Breaking multi-way ties may not be completely fair	OPLElection, IRElection	OPL3Way, likely any other places where there is an n-way tie, where $n > 2$	The way we break ties marches down a list in order, so people in the middle get more opportunities to have their ties broken than on the ends. The program still runs as intended, and a person isn't given more than one seat, but it also may not be entirely fair to the people at the ends. Not a bug per se, but something to be corrected in the future.
Quotas calculated that have decimals will be floor divided, as we use integers for our variables	OPLElection	A case where there are 4 seats and 3 ballots would create a quota of 0 due to how Java does division, and likely a litany of errors. Other scenarios like 4 seats and 5 ballots would have quotas of 1, as the decimal would be chopped	We assumed that there wouldn't be elections for more seats than there are ballots, so we don't necessarily think that this is a bug per se, but we are covering our bases. It doesn't really make sense that someone could win an election while receiving no votes, but if it's possible then that would be a bug. In the case where there are more ballots than seats and the decimal is chopped off, it was never specified which way it should be rounded, so this may not be a bug either, but we are listing it here for the

		off.	sake of completeness.
--	--	------	-----------------------