Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_AuditLog_1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates a test Audit log that prints initial ballot information, the processing of the ballot, and the results.	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** FileCreation();
Results: Pass __X___ Fail_____	`[ RUN ] fixture_AuditLog.FileCreation` `[ OK ] fixture_AuditLog.FileCreation (0 ms)`
Preconditions for Test: The audit log is requested for the program	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Print out ballot information	Log()	Ballot information can be seen		
2	Print out ballot processing	Log()	Ballot processing can be viewed		
3	Print out election results	Log()	Election results are printed		
4					

Post condition(s) for Test:

The audit file is available to be read by the auditor

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/26/2021**
Test Case ID#: test_AuditLog_2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates a test IR Ballot object and logs the ballot. The information printed to the log includes: id, list of choices, the choice index, and current choice.	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** ObjectLoggingIRBallot();
Results: Pass X Fail	`[ RUN ] fixture_AuditLog.ObjectLoggingIRBallot` `[ OK ] fixture_AuditLog.ObjectLoggingIRBallot (0 ms)`
Preconditions for Test: Audit information for and IR Ballot is requested.	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create IR Ballot	IRBallot("1,2,3,4")	IR Ballot is created/initialized		
2	Print out ballot information	Log()	Ballot information can be seen		
3					
4					

Post condition(s) for Test:

IR Ballot audit information can be seen in the audit log.

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/26/2021**
Test Case ID#: test_AuditLog_3	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates a test IR Candidate object and logs the candidate. The information printed to the log includes: their name, party, current tally count, and whether or not they have been eliminated.	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** ObjectLoggingIRCandidate();
Results: Pass __X___ Fail_____	`[ RUN ] fixture_AuditLog.ObjectLoggingIRCandidate` `[ OK ] fixture_AuditLog.ObjectLoggingIRCandidate (0 ms)`
Preconditions for Test: Audit information for IR Candidate is requested.	

Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1	Create IR Candidate	IRCandidate("Bob"," Builder")	IR Candidate is created and initialized		
2	Print out candidate information	Log()	Candidate information can be seen		
3					
4					

Post condition(s) for Test:

IR Candidate audit information is seen in the audit file

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/26/2021**
Test Case ID#: test_AuditLog_4	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates a test OPL Ballot object and logs the ballot. The information printed to the log includes the choice of the ballot.	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** ObjectLoggingOPLBallot();
Results: Pass __X___ Fail_____	`[ RUN ] fixture_AuditLog.ObjectLoggingOPLBallot` `[ OK ] fixture_AuditLog.ObjectLoggingOPLBallot (0 ms)`
Preconditions for Test: Audit information for an OPL Ballot is requested.	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL Ballot	OPLBallot(“,,1,)	Creates/initializes OPL Ballot object		
2	Print out ballot information	Log()	Ballot information can be seen		
3					
4					

Post condition(s) for Test:

OPL Ballot audit information can be seen in the audit log.

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/26/2021**
Test Case ID#: test_AuditLog_5	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates a test OPL Candidate object and logs the candidate. The information printed to the log includes the list of ballots that currently belong to that candidate.	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** ObjectLoggingOPLCandidate();
Results: Pass X Fail	`[ RUN ] fixture_AuditLog.ObjectLoggingOPLCandidate` `[ OK ] fixture_AuditLog.ObjectLoggingOPLCandidate (0 ms)`
Preconditions for Test: Audit information for an OPL Candidate is requested.	

Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1	Create OPL Candidate	OPLCandidate("bob")	OPL Candidate object is created/initialized		
2	Print out ballot information	Log()	Ballot information can be seen		
3					
4					

Post condition(s) for Test:

Audit information for an OPL Candidate can be seen in the audit log.

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/26/2021**
Test Case ID#: test_AuditLog_6	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates a test OPL Party object and logs the party. The information printed to the log includes: the name of the party, a list of candidates belonging to that party, and their ballots.	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** ObjectLoggingOPLParty();
Results: Pass X Fail	`[ RUN ] fixture_AuditLog.ObjectLoggingOPLParty` `[ OK ] fixture_AuditLog.ObjectLoggingOPLParty (0 ms)`
Preconditions for Test: Audit information for an OPL Party is requested.	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL Party	OPLParty (“builder”)	OPL Party object is created/initialized		
2	Print out ballot information	Log()	Ballot information can be seen		
3					
4					

Post condition(s) for Test:

Audit information for the OPL Party can be seen in the audit log.

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test Ballot 1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates a unique id for each ballot	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** BallotUniqueID

Results: Pass X Fail	`[----------] 2 tests from fixture_Ballot`
	`[ RUN ] fixture_Ballot.BallotUniqueID`
	`[ OK ] fixture_Ballot.BallotUniqueID (0 ms)`
	`[ RUN ] fixture_Ballot.BallotUniqueIDRepeat`
	`[ OK ] fixture_Ballot.BallotUniqueIDRepeat (78 ms)`
	`[----------] 2 tests from fixture_Ballot (78 ms total)`

Preconditions for Test: Test ballot is created	

Step	Test Step	Test	Expected	Actual	

#	Description	Data	Result	Result	Notes
1	create test ballot	TestBallot()	Ballot is created		
2	create a unique id for test balot	get_id()	Ballot receives an id		
3					
4					

Post condition(s) for Test:

The ballot can now be referenced by other parts of the program given ID

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_Ballot_2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates database for all ballots after they have an id and generates a lot of ballots to test and make sure each id is unique	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** BallotUniqueIDRepeat

Results: Pass __X_____ Fail _____	`[----------] 2 tests from fixture_Ballot`
	`[ RUN ] fixture_Ballot.BallotUniqueID`
	`[ OK ] fixture_Ballot.BallotUniqueID (0 ms)`
	`[ RUN ] fixture_Ballot.BallotUniqueIDRepeat`

	[OK] fixture_Ballot.BallotUniqueIDRepeat (78 ms) [----------] 2 tests from fixture_Ballot (78 ms total)
Preconditions for Test: Each ballot has an assigned id	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	store ballots in a database	std::set<uint64_t> ids	Ballot id's are now stored in a database		
2	generate many ballots	TestBallot()	Many ballots are generated		
3	check to make sure generated and stored ballots are unique	get_id()	all ballots have unique id's		
4					

Post condition(s) for Test:

All ballots created have a unique id

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/27/2021**
Test Case ID#: test_ElectionFactory_1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Passing a IR ballot file to the ElectionFactory produces an IRElection	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** test_ElectionFactory.cc,

	fixture_ElectionFactory.ElectionFactoryIR
Results: Pass __X___ Fail_____	`[ RUN ] fixture_ElectionFactory.ElectionFactoryIR [ OK ] fixture_ElectionFactory.ElectionFactoryIR (1 ms)`
Preconditions for Test: An IR ballot file exists	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Pass filename to election factory	std::string filename	Factory finds file	Found file	
2	Get returned election type	Election::get_type()	returns "IR"	returned "IR"	
3					
4					

Post condition(s) for Test:
A new election is created with an IR type

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/27/2021**
Test Case ID#: test_ElectionFactory_2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Passing a OPL ballot file to the ElectionFactory produces an OPLElection	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.**

	test_ElectionFactory.cc, fixture_ElectionFactory.ElectionFactoryOPL

Results: Pass __X___ Fail_____	`[ RUN ] fixture_ElectionFactory.ElectionFactoryOPL`
	`D11`
	`D21`
	`R12`
	`R22`
	`R32`
	`I13`
	`[ OK ] fixture_ElectionFactory.ElectionFactoryOPL (1 ms)`

Preconditions for Test: An OPL ballot file exists	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Pass filename to election factory	std::string filename	Factory finds file	Found file	
2	Get returned election type	Election::get_type()	returns "OPL"	returned "OPL"	
3					
4					

Post condition(s) for Test:
A new election is created with an OPL type

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_ElectionFactory_3	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**

Test Description: Passing a non-existent file to the ElectionFactory produces a nullptr	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** test_ElectionFactory.cc, fixture_ElectionFactory.ElectionFactoryNotFound

Results: Pass X Fail	`[ RUN ] fixture_ElectionFactory.ElectionFactoryNotFound`
	`Failed to open non_existent.csv`
	`Reason: No such file or directory`
	`[ OK ] fixture_ElectionFactory.ElectionFactoryNotFound (0 ms)`

Preconditions for Test: A specified ballot file does not exist	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Pass filename to election factory	std::string filename	Factory finds file	Found file	
2	Test election pointer is null	Election*	Is nullptr	Is nullptr	
3					
4					

Post condition(s) for Test:
No new election is created

Project Name: Project 1: Voting System	**Team# 20**

Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_IRBallot_1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests the basic functionality of the IRBallotCSVParse function	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRBallotCSVParse()

Results: Pass ___X__ Fail_____	`[----------] 3 tests from fixture_IRBallot`
	`[ RUN ] fixture_IRBallot.IRBallotCSVParse`
	`[ OK ] fixture_IRBallot.IRBallotCSVParse (0 ms)`
	`[ RUN ] fixture_IRBallot.IRBallotIncrement`
	`[ OK ] fixture_IRBallot.IRBallotIncrement (0 ms)`
	`[ RUN ] fixture_IRBallot.IRBallotLog`
	`[ OK ] fixture_IRBallot.IRBallotLog (0 ms)`
	`[----------] 3 tests from fixture_IRBallot (0 ms total)`

Preconditions for Test: Ballots have been created	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	create a new ballot with election data and store it	IRBallot(1,2,3,4)	ballot information is stored correctly		
2	extract the ballot information	get_choice()	ballot information has been read correctly and prints correctly		
3					
4					

Post condition(s) for Test:

Ballot information can now be read and used by other parts of the code.

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_IRBallot_2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests the basic functionality of the IRBallotIncrement function	
Automated: yes X no __	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRBallotIncrement()

Results: Pass X Fail	`[----------] 3 tests from fixture_IRBallot`
	`[ RUN ] fixture_IRBallot.IRBallotCSVParse`
	`[ OK ] fixture_IRBallot.IRBallotCSVParse (0 ms)`
	`[ RUN ] fixture_IRBallot.IRBallotIncrement`
	`[ OK ] fixture_IRBallot.IRBallotIncrement (0 ms)`
	`[ RUN ] fixture_IRBallot.IRBallotLog`
	`[ OK ] fixture_IRBallot.IRBallotLog (0 ms)`
	`[----------] 3 tests from fixture_IRBallot (0 ms total)`

Preconditions for Test: Ballots have been created	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	create a new ballot with election data and store it	IRBallot(1,2,3,4)	ballot information is stored correctly		
2	extract the ballot information and increment the choice based on the ballot information	increment_choice();	ballot information has been read correctly and prints correctly		
3					
4					

Post condition(s) for Test:

Ballots can now be accessed by other parts of the code via the created database

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/14/2021**
Test Case ID#: test_IRBallot_3	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests the basic functionality of the IRBallotLog function	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRBallotLog()

Results: Pass__X____ Fail_____	`[----------] 3 tests from fixture_IRBallot`
	`[ RUN ]`
	`fixture_IRBallot.IRBallotCSVParse`
	`[ OK ]`
	`fixture_IRBallot.IRBallotCSVParse (0 ms)`
	`[ RUN ]`
	`fixture_IRBallot.IRBallotIncrement`

	[OK] fixture_IRBallot.IRBallotIncrement (0 ms) [RUN] fixture_IRBallot.IRBallotLog [OK] fixture_IRBallot.IRBallotLog (0 ms) [----------] 3 tests from fixture_IRBallot (0 ms total)
Preconditions for Test: Ballots have been created	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	create a new ballot with election data and store it	IRBallot(1,2,3,4)	ballot information is stored correctly		
2	stores ballot information	get_id()	ballot information has been read correctly and prints correctly		
3	takes stored ballot information and makes it accessible to log to the audit file	log()the	ballot information is printed to audit log		
4					

Post condition(s) for Test:

Audit log prints ballot information

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/14/2021**
Test Case ID#: test IRBallot 4	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**

Test Description: Tests that the IRBallot constructor handles ballot choices in different orders	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** test_IRBallot.cc, IRBallot::IRBallot()
Results: Pass X Fail	`[ RUN ] fixture_IRBallot.IRBallotOrder` `[ OK ] fixture_IRBallot.IRBallotOrder (0 ms)`
Preconditions for Test: Strings with different ballot preferences exist	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Construct a IRBallot	Ballot choice string			Previously tested
2	Test choice order is correct		Order matches	Order matches	
3					
4					

Post condition(s) for Test:

IRBallots are created with the proper choice order

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_IRCandidate_1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**

Test Description: Tests the IRCandidate constructor and verifies the name, party, and other structures are properly set.	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** test_IRCandidate.cc, IRCandidate::IRCandidate()
Results: Pass X Fail	`[ RUN ] fixture_IRCandidate.IRCandidateConstructor` `[ OK ] fixture_IRCandidate.IRCandidateConstructor (0 ms)`
Preconditions for Test: A candidate name and party has been selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Make a new IRCandidate	Candidate name, party	Object is created	Object is created	
2	Test candidate name	Candidate name	Name matches	Name matches	
3	Test candidate party	Candidate party	Party matches	Party matches	
4	Test number of ballots	get_tally()	Zero ballots	Zero ballots	

Post condition(s) for Test:

A new IRCandidate is created

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/27/2021**
Test Case ID#: test_IRCandidate_2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**

Test Description: Tests the basic functionality of the IRCandidate::add_ballot() method	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** test_IRCandidate.cc, IRCandidate::add_ballot()
Results: Pass X Fail	**[RUN] fixture_IRCandidate.IRCandidateAddBallot [OK] fixture_IRCandidate.IRCandidateAddBallot (0 ms)**
Preconditions for Test: IRBallots can be created	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Make IRCandidate		Ok	Ok	Previously tested
2	Add ballots	IRBallots	Candidate tally increases by proper amount	Tally increases by the added amount	
3					
4					

Post condition(s) for Test:

IRCandidate's tally is increased

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/27/2021**
Test Case ID#: test_IRCandidate_3	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**

Test Description: Tests the basic functionality of the IRCandidate::eliminate() function	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** test_IRCandidate.cc, IRCandidate::eliminate()
Results: Pass_____ Fail__X____	`[ RUN ]` **`fixture_IRCandidate.IRCandidateEliminate`** **`/home/alex/UMN/csci5801/repo-Team20/Project1/testing/test_IRCandidate.cc:124: Failure`** **`Failed`** **`IRCandidate should thow exception on a second elimination attempt`** **`[ FAILED ]`** **`fixture_IRCandidate.IRCandidateEliminate (0 ms)`**
Preconditions for Test: IRCandidates can be constructed	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Make IRCandidate		Ok	Ok	Previously tested
2	Eliminate candidate		Candidate eliminated	Candidate eliminated	
3	Eliminate candidate		Candidate still eliminated, exception thrown	Candidate still eliminated	Function works, but does not safeguard against multiple calls
4					

Post condition(s) for Test:

IRCandidate has been eliminated

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/27/2021**
Test Case ID#: test_IRCandidate_4	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests the basic functionality of the IRCandidate::clear_ballots() function	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** test_IRCandidate.cc, IRCandidate::clear_ballots()
Results: Pass Fail X	`[ RUN ] fixture_IRCandidate.IRCandidateClearBallots /home/alex/UMN/csci5801/repo-Team20/Project1 /testing/test_IRCandidate.cc:155: Failure Failed Clearing ballots before elimination should throw exception [ FAILED ] fixture_IRCandidate.IRCandidateClearBallots (0 ms)`
Preconditions for Test: IRCandidates can be constructed and eliminated	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Make IRCandidate		Ok	Ok	Previously tested
2	Eliminate candidate				Previously tested
3	Clear ballots from candidate		No ballots in candidate	No ballots remaining	
4	Make IRCandidate		Ok	Ok	Previously tested
	Clear ballots from candidate		No ballots in candidate, exception throw because we didn't eliminate candidate	No ballots remaining, no exception	Function works, but does not safeguard against improper usage

Post condition(s) for Test:

IRCandidate has ballots cleared

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/27/2021**
Test Case ID#: test IRCandidate 5	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests the basic functionality of the IRCandidate::log() function	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** test_IRCandidate.cc, IRCandidate::log()
Results: Pass __X___ Fail_____	`[ RUN ] fixture_IRCandidate.IRCandidateLog` `[ OK ] fixture_IRCandidate.IRCandidateLog (0 ms)`
Preconditions for Test: IRCandidates can be constructed, ballots added, and eliminated	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Make a IRCandidate				Previously tested
2	Add ballots to candidate				Previously tested
3	Get the log string		All relevant fields are contained in the string	All relevant fields are contained in the string	
4					

Post condition(s) for Test:

IRCandidate produces necessary information in the log string

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System _X_	**Test Date: 03/14/2021**
Test Case ID#: test_IRElection_1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**

Test Description:
 	
Tests if the IRElection runs properly
	`[ RUN ] fixture_IRElection.IRElectionSRSExample`
	`The winner of the election is Rosen of party D with 66.6667% of the votes`
	`Any remaining candidates were eliminated.`
	`[ OK ] fixture_IRElection.IRElectionSRSExample (0 ms)`

Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRElection()
Results: Pass X Fail	
Preconditions for Test: IR Election has been selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	start new IRElection	IRElection()	new IR Election has been started		
2	run the IR Election	run();	IR Election runs without any errors		
3					
4					

Post condition(s) for Test:

Election results can be viewed from IR Election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System _X_	**Test Date: 03/14/2021**
Test Case ID#: test IRElection 2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests if the IRElection runs properly with random values/candidates	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRElection()

Results: Pass __X___ Fail_____	`[ RUN ]`
	`fixture_IRElection.IRElectionRandom`
	`The winner of the election is Candidate6Test of party 6 with 50.222% of the votes`
	`Any remaining candidates were eliminated.`
	`[ OK ]`
	`fixture_IRElection.IRElectionRandom (1268 ms)`

Preconditions for Test: IR Election has been selected	

Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1	start new IRElection	IRElection()	new IR Election has been started		
2	run the IR Election	run();	IR Election runs without any errors		
3					
4					

Post condition(s) for Test:

Election results can be viewed from IR Election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System _X_	**Test Date: 03/14/2021**
Test Case ID#: test_IRElection_3	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests if the IRElection runs properly with random values/candidates	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRElection()

Results: Pass __X___ Fail _____	`[ RUN ]`
	`fixture_IRElection.IRElectionMPV`
	`The winner of the election is Royce of party L with 114.286% of the votes`
	`Any remaining candidates were eliminated.`
	`[ OK ]`
	`fixture_IRElection.IRElectionMPV (0 ms)`

Preconditions for Test: IR Election has been selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	start new IRElection	IRElection()	new IR Election has been started		
2	run the IR Election	run();	IR Election runs without any errors	No errors	
3					
4					

Post condition(s) for Test:

Election results can be viewed from IR Election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System _X_	**Test Date: 03/14/2021**
Test Case ID#: test_IRElection_4	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests if the IRElection runs properly with random values/candidates	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRElection()

Results: Pass__X___ Fail_____	`[ RUN ] fixture_IRElection.IRElectionNM`
	`The winner of the election is Royce of party L with 100% of the votes`
	`Any remaining candidates were eliminated.`
	`[ OK ] fixture_IRElection.IRElectionNM (0 ms)`

Preconditions for Test: IR Election has been selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	start new IRElection	IRElection()	new IR Election has been started		
2	run the IR Election	run();	IR Election runs without any errors	No errors	
3					
4					

Post condition(s) for Test:

Election results can be viewed from IR Election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System _X_	**Test Date: 03/14/2021**
Test Case ID#: test_IRElection_5	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests if the IRElection runs properly with random values/candidates	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRElection()

Results: Pass__X___ Fail_____	`[ RUN ] fixture_IRElection.IRElectionOC`
	`The winner of the election is Rosen of party D with 100% of the votes`
	`Any remaining candidates were eliminated.`
	`[ OK ] fixture_IRElection.IRElectionOC (9 ms)`

Preconditions for Test: IR Election has been selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	start new IRElection	IRElection()	new IR Election has been started		
2	run the IR Election	run();	IR Election runs without any errors	No errors	
3					
4					

Post condition(s) for Test:

Election results can be viewed from IR Election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System _X_	**Test Date: 03/14/2021**
Test Case ID#: test_IRElection_6	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests if the IRElection runs properly with random values/candidates	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRElection()

Results: Pass__X___ Fail_____	`[ RUN ] fixture_IRElection.IRElectionSV`
	`The winner of the election is Royce of party L with 100% of the votes`
	`Any remaining candidates were eliminated.`
	`[ OK ] fixture_IRElection.IRElectionSV (0 ms)`

Preconditions for Test: IR Election has been selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	start new IRElection	IRElection()	new IR Election has been started		
2	run the IR Election	run();	IR Election runs without any errors	No errors	
3					
4					

Post condition(s) for Test:

Election results can be viewed from IR Election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System _X_	**Test Date: 03/14/2021**
Test Case ID#: test_IRElection_7	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests if the IRElection runs properly with random values/candidates	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** IRElection()

Results: Pass __X___ Fail_____	`[ RUN ]`
	`fixture_IRElection.IRElectionTWT`
	`The winner of the election is Royce of party L with 70% of the votes`
	`Any remaining candidates were eliminated.`
	`[ OK ]`
	`fixture_IRElection.IRElectionTWT (0 ms)`

Preconditions for Test: IR Election has been selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	start new IRElection	IRElection()	new IR Election has been started		
2	run the IR Election	run();	IR Election runs without any errors	No errors	
3					
4					

Post condition(s) for Test:

Election results can be viewed from IR Election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_MediaReport_FileCreation	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests if information from the election and audit log can be printed in a media report	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** MediaReport()

Results: Pass X Fail	`[----------] 1 test from fixture_MediaReport`
	`[ RUN ] fixture_MediaReport.MediaReportFileCreation`
	`[ OK ] fixture_MediaReport.MediaReportFileCreation (1 ms)`
	`[----------] 1 test from fixture_MediaReport (1 ms total)`

Preconditions for Test: The election has finished	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	create media report	MediaReport()	A media		
2	write to the media report	write()	election results have been written to the media report and are		

			correct		
3					
4					

Post condition(s) for Test:

Election results have been shared with the media

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/26/2021**
Test Case ID#: test_MediaReport_write	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Tests if information from the election and audit log can be written to a media report	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** MediaReport()
Results: Pass __X___ Fail_____	`[ RUN ] fixture_MediaReport.write` `[ OK ] fixture_MediaReport.write (0 ms)`
Preconditions for Test: A string is ready to be written to the report	

Step	**Test Step**	**Test**	**Expected**	**Actual**	

#	Description	Data	Result	Result	Notes
1	create media report	MediaReport()	A media report was created	A media report was created	
2	write to the media report	write()	Specified strings were written to the report	The strings were written to the report properly	
3					
4					

Post condition(s) for Test:
A media report is ready for distribution

Election results have been shared with the media

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System _	**Test Date: 03/25/2021**
Test Case ID#: test_OPLBallot_1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL ballot that is a CSV and parses a single choice from the ballot	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLBallot()
Results: Pass _X_ Fail _	`[ RUN ] fixture_OPLBallot.OPLBallotCSVParse_1choice` `[ OK ] fixture_OPLBallot.OPLBallotCSVParse_1choice (0 ms)`
Preconditions for Test: OPL election is selected	

Step	Test Step	Test	Expected	Actual	

#	Description	Data	Result	Result	Notes
1	Create OPL Ballot	OPLBallot()	OPL Ballot has been created		
2	read information from ballot	get_choice()	ballot information has been read		
3					
4					

Post condition(s) for Test:

Ballot information can be viewed and used by other parts of the code

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_OPLBallot_2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL ballot that is a CSV and parses the file with no choices from the ballot	`[ RUN ] fixture_OPLBallot.OPLBallotCSVParse_0choice` `[ OK ] fixture_OPLBallot.OPLBallotCSVParse_0choice (0 ms)`
Automated: _yes_X_ no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLBallot()
Results: Pass __X___ Fail_____	
Preconditions for Test: OPL election is selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL Ballot	OPLBallot()	OPL Ballot has been created		
2	read information from ballot	get_choice()	ballot information has been read		

3					
4					

Post condition(s) for Test:

Ballot information can be viewed and used by other parts of the code

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_OPLBallot_3	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL ballot and logs a single choice from the ballot	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLBallot()
Results: Pass __X___ Fail_____	`[ RUN ] fixture_OPLBallot.OPLBallotLogOnceChoice` `[ OK ] fixture_OPLBallot.OPLBallotLogOnceChoice (0 ms)`
Preconditions for Test: OPL election is selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL Ballot	OPLBallot()	OPL Ballot has been created		
2	retrieve ballot with information	get_id()	ballot has been selected and information has been read		
3	log ballot infromation	log()	ballot information can be viewed in the audit log		

4					

Post condition(s) for Test:

Audit log results are printed

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/14/2021**
Test Case ID#: test_OPLBallot_4	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL ballot and logs a no choices from the ballot	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLBallot()
Results: Pass X Fail	**[RUN] fixture_OPLParty.OPLParty_1party_1candidate** **[OK] fixture_OPLParty.OPLParty_1party_1candidate (0 ms)**
Preconditions for Test: OPL election is selected	

Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1	Create OPL Ballot	OPLBallot()	OPL Ballot has been created		
2	retrieve ballot with information	get_id()	ballot has been selected and information has been read		
3	log ballot infromation	log()	ballot information can be viewed in the audit log		

4					

Post condition(s) for Test:

Audit log results are printed

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_OPLCandidate_1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an empty OPLCandidate	
Automated: yes X no __	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLCandidate()
Results: Pass __X__ Fail __	**[RUN] fixture_OPLCandidate.OPLCandidate_empty_log** **[OK] fixture_OPLCandidate.OPLCandidate_empty_log (0 ms)**
Preconditions for Test: OPL election is selected	

Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1	Create OPL Candidate	OPLCandidate()	OPL Candidate has been created		
2	log candidate information	log()	Candidate information can be viewed from the audit log - expected to be of the form: "Candidate <X>: ballots = []"		

Post condition(s) for Test:

Audit log results are printed

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/14/2021**
Test Case ID#: test_OPLCandidate_2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL candidate with no votes	
Automated: yes X no	**Indicate where you are storing the tests (what file) and the name of the method/functions being used.** OPLCandidate
Results: Pass X Fail	**[RUN]** **fixture_OPLCandidate.OPLCandidate_empty_gettally** **[OK]** **fixture_OPLCandidate.OPLCandidate_empty_gettally (0 ms)**
Preconditions for Test: OPL election is selected	

Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1	Create OPL candidate	OPLCandidate()	OPL candidate has been created		
2	retrieve the number of votes given to a candidate	get_tally()	the number of votes that have been read for a particular candidate - 0	0	
3					
4					

Post condition(s) for Test:

Candidate wins election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/14/2021**
Test Case ID#: test_OPLCandidate_3	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL candidate who has votes	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLCandidate
Results: Pass X Fail	**[RUN] fixture_OPLCandidate.OPLCandidate_gettally** **[OK] fixture_OPLCandidate.OPLCandidate_gettally (0 ms)**
Preconditions for Test: OPL election is selected	

Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1	Create OPL candidate	OPLCandidate()	OPL candidate has been created		
2	retrieve the number of votes given to a candidate	log()	the number of votes that have been read for a particular candidate		
3					
4					

Post condition(s) for Test:

Candidate wins election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_OPL_Party_1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**

Test Description: Creates an OPL party with candidates in the party	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLParty()
Results: Pass __X___ Fail_____	**[RUN] fixture_OPLParty.OPLParty_getname** **[OK] fixture_OPLParty.OPLParty_getname (0 ms)**
Preconditions for Test: OPL election is selected	

Step	**Test Step**	**Test**	**Expected**	**Actual**	

#	Description	Data	Result	Result	Notes
1	Create OPL party	OPLParty()	OPL party has been created		
2	retrieve the names of the candidates in the party	get_name()	the name of the candidates in the party has been received		
3					
4					

Post condition(s) for Test:

Candidates party wins the election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_OPL_Party_2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL party with candidates in the party and votes for each candidate and party	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLParty()
Results: Pass __X___ Fail_____	**[RUN] fixture_OPLParty.OPLParty_gettally** **[OK] fixture_OPLParty.OPLParty_gettally (0 ms)**

Preconditions for Test:

OPL election is selected

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL party	OPLParty()	OPL party has been created		
2	retrieve votes of each candidate in a party	get_tally()	votes for each candidate have been read		
3					
4					

Post condition(s) for Test:

Candidates party wins the election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_OPL_Party_3	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL party with candidates in the party and votes for each candidate and party	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLParty()

Results: Pass __X___ Fail_____	**[RUN] fixture_OPLParty.OPLParty_log** **[OK] fixture_OPLParty.OPLParty_log (0 ms)**
Preconditions for Test: OPL election is selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL party	OPLParty()	OPL party has been created		
2	Retrieve record of candidates and ballots associated with party	log()	Votes for specific candidates in a party are received and displayed	“OPL Party Democrat: []”	
3					

4					

Post condition(s) for Test:

Candidates party wins the election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_OPL_Party_4	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**

Test Description: Creates an OPL party with candidates in the party	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLParty()
Results: Pass __X___ Fail_____	**[RUN] fixture_OPLParty.OPLParty_1party_1candidate** **[OK] fixture_OPLParty.OPLParty_1party_1candidate (0 ms)**
Preconditions for Test: OPL election is selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL party	OPLParty()	OPL party has been created		
2	retrieve the votes of the candidates in the party	get_tally()	the name of the candidates in the party has been received		
3					
4					

Post condition(s) for Test:

Candidates party wins the election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_OPL_Party_5	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL party with 1 candidate, 1 party, and 3 ballots	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLParty()
Results: Pass __X___ Fail_____	**[RUN] fixture_OPLParty.OPLParty_1party_1candidate_3ballots** **[OK] fixture_OPLParty.OPLParty_1party_1candidate_3ballots (0 ms)**

Preconditions for Test: OPL election is selected					
Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1	Create OPL party	OPLParty()	OPL party has been created		
2	Create 3 new OPLBallot() instances	OPLBallot()	3 OPLBallots have been created	As expected	
3	Create OPLCandidate and add ballots to OPLCandidate	OPLCandidate() and OPLCandidate::add_ballot()	OPLCandidate has been created, and 3 ballots have been added to their stockpile	As expected	
4	Add candidate to the party	OPLParty::add_candidate(candidate)	the name of the candidates in the party has been received	As expected	
5	retrieve the votes of the candidates in the party	get_tally()	the votes of the candidates in the party has been counted and displayed, 3 should be expected	3 votes are the result of ->get_tally()	

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_OPL_Party_6	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL party with 4 candidates and 6 ballots spread between them, and calls get_top_n_candidates(4) to ensure that the method properly sorts them based on vote getting	

Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLParty()
Results: Pass __X___ Fail_____	**[RUN] fixture_OPLParty.OPLParty_1party_1candidate_3ballots** **[OK] fixture_OPLParty.OPLParty_1party_1candidate_3ballots (0 ms)**

Preconditions for Test:

OPL election is selected

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL party	OPLParty()	OPL party has been created	Yes	
2	Create 4 OPL Candidates	OPLCandidate()	4 unique candidates have been created	Yes	

3	Create 6 ballots, with 3 / 2 / 1 / 0 voting for candidate 1 /2 /3 /4, respectively, and adding them to each candidate's vector	OPLBallot(,,,), OPLCandidate::add_ballot()	Each candidate has the correct number of ballots	Yes	
4	Call get_top_n_candidates(4) to get a list ordered by votes	OPLParty::get_top_n_candidates(n)	A correct sorted list is displayed	Yes	Noted in buglist, but occasionally this method runs into a segfault

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/25/2021**

Test Case ID#: test_OPL_Party_7	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL party with 2 candidates with 1 vote each, and calls get_top_n_candidates(1), indicating that only the top candidate should get selected. This will test the ability for the tiebreaker method to adequately and equally select either candidate randomly.	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLParty()
Results: Pass __X___ Fail_____	**[RUN] fixture_OPLParty.OPLParty_tiebreaker** **[OK] fixture_OPLParty.OPLParty_tiebreaker (41626 ms)**

Preconditions for Test:

OPL election is selected, candidates are added to party, and votes are allocated to candidates

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL party	OPLParty()	OPL party has been created	As expected	
2	create new candidate	OPLCandidate()	name of the candidate is read	As expected	
3	add the candidate to a party	add_candidate()	candidate has been added to a party	As expected	
4	create ballots with voter information	OPLBallot()	ballots have been created	As expected	
5	add ballots to a specific candidate based on	add_ballot()	ballots have been correctly assigned to the candidate	As expected	
	Call OPLParty:get_top_n_candidates	get_top_n_candidates	Round(10*(avg/1000000) = 5 – Each candidate will be selected as the top candidate an approximately equal number of times over the course of 100,000 repeats (that is, the probability of candidate 1 being selected is 5/10 = 0.5	As expected, 0.5 = avg	

Post condition(s) for Test:

The proportion of total tiebreaker tests which results in 1 out of 2 candidates being chosen has been determined to be 0.5.

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test OPL Election 1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL party with 1 candidate, 1 party, and 3 ballots	
Automated: yes_X__ no ___	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLParty()
Results: Pass X Fail	**[RUN] fixture_OPLElection.OPLElectionFunctionality** **[OK] fixture_OPLElection.OPLElectionFunctionality (722 ms)**
Preconditions for Test: OPL election is selected	

Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1	Create OPL party	OPLParty()	OPL party has been created		
2	create new candidate	OPLCandidate()	name of the candidate is read		
3	add the candidate to a party	add_candidate()	candidate has been added to a party		
4	create ballots with voter information	OPLBallot()	information		
5	add ballots to a specific cand				

Post condition(s) for Test:

Candidates party wins the election

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit X System	**Test Date: 03/14/2021**
Test Case ID#: test OPLElection 2	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Creates an OPL Election object to run on a ballot csv file where there is a three way tie between the 3 parties	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** OPLElection()

Results: Pass Fail X	`[ RUN ]`
	`fixture_OPLBallot.OPLBallotCSVParse_1choice`
	`segmentation fault`

Preconditions for Test: OPL election is selected	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create OPL Election	OPLElection()	OPL Election has been created		
2	Run overall structure of OPLElection	run()	Ballot file has been read and results have been compiled and outputted	Segmentation fault!	
3					
4					

Post condition(s) for Test:

All votes have been recorded and counted

Project Name: Project 1: Voting System	**Team# 20**
Test Stage: Unit _X_ System __	**Test Date: 03/14/2021**
Test Case ID#: test_TieBreaker_1	**Name(s) of Testers: Alex, Nikhil, Peter, Andrew**
Test Description: Checks to see if the tie breaker function is functional in resolving ties	
Automated: yes X no	**Indicate where are you storing the tests (what file) and the name of the method/functions being used.** TieBreaker()
Results: Pass X Fail	`[ RUN ] fixture_TieBreaker.TieBreakerInRange` `[ OK ] fixture_TieBreaker.TieBreakerInRange (425 ms)`
Preconditions for Test: A tie in the election occurs	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	creates a loop to go through coin flips and check to see if the function can resolve the tie properly	resolve_tie()	function resolves tie and assigns winner		
2					
3					
4					
5					

Post condition(s) for Test:

Winner is announced