Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __
Test Case ID#: EDOVD_1a
Test Description: This will test
if the method verifyLineIsDigit(String line)
correctly verify if the strings with only numbers
pass

Test Date: 3/25/24

Name(s) of Testers: Bethany Freeman

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest.java
verifyLineIsDigit()

Automated: yes x no ___

Results: Pass ____x_ Fail_____

Preconditions for Test: An ExtractDataOPL object must have been created

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new ExtractDataOPL object	ExtractDataOPL test01= new ExtractDataOPL(validFile01, "OPL")	Object created	Object created	
3	Create a String line	String line = "";	variable created	variable created	
4	assert that false is given when test01.verifyLineIsDigit(line) is called	false, test01.verifyLineIsDigit(line)	true	true	

Post condition(s) for Test:
VerifyLineIsDigit(String) has been verified to not work with an empty string

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __
Test Case ID#: EDOVD_1b
Test Description: This will test if the method verifyLineIsDigit(String line) correctly verify if the strings with only numbers pass

Test Date: 3/25/24

Name(s) of Testers: Bethany Freeman

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest.java
verifyLineIsDigit()

Automated: yes x no ___

Results: Pass ____x_ Fail_____

Preconditions for Test: An ExtractDataOPL object must have been created

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new ExtractDataOPL object	ExtractDataOPL test01= new ExtractDataOPL(validFile01, "OPL")	Object created	Object created	
3	Create a String line	String line = "1";	variable created	variable created	
4	assert that true is given when test01.verifyLineIsDigit(line) is called	true, test01.verifyLineIsDigit(line)	true	true	

Post condition(s) for Test:
VerifyLineIsDigit(String) has been verified to work with a digit

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __
Test Date: 3/25/24
Test Case ID#: EDOVD_1c
Test Description: This will test
if the method verifyLineIsDigit(String line)
correctly verify if the strings with only numbers
pass

Name(s) of Testers: Bethany Freeman

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest.java
verifyLineIsDigit()

Automated: yes x no ___

Results: Pass ____x_ Fail_____

Preconditions for Test: An ExtractDataOPL object must have been created

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new ExtractDataOPL object	ExtractDataOPL test01= new ExtractDataOPL(validFile01, "OPL")	Object created	Object created	
3	Create a String line	String line = " 1"	variable created	variable created	
4	assert that true is given when test01.verifyLineIsDigit(line) is called	true, test01.verifyLineIsDigit(line)	true	true	

Post condition(s) for Test:
VerifyLineIsDigit(String) has been verified to work with a digit and space

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __
Test Case ID#: EDOVD_1d
Test Description: This will test
if the method verifyLineIsDigit(String line)
correctly verify if the strings with only numbers
pass

Test Date: 3/25/24

Name(s) of Testers: Bethany Freeman

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest.java
verifyLineIsDigit()

Automated: yes x no ___

Results: Pass ____x_ Fail_____

Preconditions for Test: An ExtractDataOPL object must have been created

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new ExtractDataOPL object	ExtractDataOPL test01= new ExtractDataOPL(validFile01, "OPL")	Object created	Object created	
3	Create a String line	String line = "abc";	variable created	variable created	
4	assert that false is given when test01.verifyLineIsDigit(line) is called	false, test01.verifyLineIsDigit(line)	true	true	

Post condition(s) for Test:
VerifyLineIsDigit(String) has been verified to not work with a letter string

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __
Test Case ID#: EDOVD_1e
Test Description: This will test
if the method verifyLineIsDigit(String line)
correctly verify if the strings with only numbers
pass

Test Date: 3/25/24

Name(s) of Testers: Bethany Freeman

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest.java
verifyLineIsDigit()

Automated: yes x no ___

Results: Pass ____x_ Fail_____

Preconditions for Test: An ExtractDataOPL object must have been created

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new ExtractDataOPL object	ExtractDataOPL test01= new ExtractDataOPL(validFile01, "OPL")	Object created	Object created	
3	Create a String line	String line = "12b";	variable created	variable created	
4	assert that false is given when test01.verifyLineIsDigit(line) is called	false, test01.verifyLineIsDigit(line)	true	true	

Post condition(s) for Test:
VerifyLineIsDigit(String) has been verified to not work with a letter and digit string

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __

Test Date: 3/26/24

Test Case ID#: EDOFP_2a

Test Description: Format parties into partyCandidates with the correct number of parties passed in

Name(s) of Testers: Bethany Freeman

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest,java
formatPartyInformation()

Automated: yes__x_ no ___

Results: Pass ____ x ____ Fail _____

Preconditions for Test: an initialized partyVotes and candidateVotes, empty. a file that exists. an ExtractDataOPL Object

Step # 1	Test Step Description	Test Data	Expected Result	Actual Result	Notes
2	Create a new BufferedReader Object	validFile = new BufferedReader(new FileReader(new File("src/test/java/InputFiles/OPLPartyInfo01.txt")))	Object Created	Object Created	
3	Create ExtractDataOPL Object	test01 = new ExtractDataOPL(validFile, "OPL");	Object Created	Object Created	
4	Call formatPartyInformation	partyCandidates = test01.formatPartyInformation(5, partyVotes, candidateVotes);	Object instantiated	Object Instantiated	
5	Create HashMap<String, ArrayList<String>>	HashMap<String, ArrayList<String>> expected = new HashMap<String, ArrayList<String>>(); expected.put("Pluto", new ArrayList<>(Arrays.asList(" Becky", " Mariah"))); expected.put("Green", new ArrayList<>(Arrays.asList(" Jonah", " Radius", " Louis")));	Object Created	Object Created	
6	Assert Expected equals party candidates	expected, partyCandidates	true	true	

Post condition(s) for Test:
formatPartyInformation() given the correct number of parties will correctly format the parties into a Hashmap

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __

Test Date: 3/26/24

Test Case ID#: EDOFP_2b

Test Description: Format parties into partyCandidates with the incorrect number of parties passed in

Name(s) of Testers: Bethany Freeman

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest,java
formatPartyInformation()

Automated: yes__x_ no ___

Results: Pass ____ x ____ Fail _____

Preconditions for Test: an initialized partyVotes and candidateVotes, empty. a file that exists. an ExtractDataOPL Object

Step # 1	Test Step Description	Test Data	Expected Result	Actual Result	Notes
2	Create a new BufferedReader Object	validFile = new BufferedReader(new FileReader(new File("src/test/java/InputFiles/OPLPartyInfo01.txt")))	Object Created	Object Created	
3	Create ExtractDataOPL Object	test01 = new ExtractDataOPL(validFile, "OPL");	Object Created	Object Created	
4	Call formatPartyInformation	partyCandidates = test01.formatPartyInformation(0, partyVotes, candidateVotes);	Object instantiated	Object Instantiated	
5	Create HashMap<String, ArrayList<String>>	expected = new HashMap<String, ArrayList<String>>();	Object Created	Object Created	
6	Assert Expected equals party candidates	expected, partyCandidates	true	true	

Post condition(s) for Test:
formatPartyInformation() given the correct number of parties will correctly format the parties into a Hashmap

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __

Test Date: 3/26/24

Test Case ID#: EDOFP_2c

Name(s) of Testers: Bethany Freeman

Test Description: Format parties into partyCandidates where a party has no candidates

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest,java
formatPartyInformation()

Automated: yes__x_ no ___

Results: Pass ____ x ____ Fail ____

Preconditions for Test: an initialized partyVotes and candidateVotes, empty. a file that exists. an ExtractDataOPL Object

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new BufferedReader Object	validFile = new BufferedReader(new FileReader(new File("src\\test\\java\\InputFiles\\OPLPartyInfo02.txt")))	Object Created	Object Created	
3	Create ExtractDataOPL Object	test01 = new ExtractDataOPL(validFile, "OPL");	Object Created	Object Created	
6	Assert formatPartyInformation throws IOException	IOException.class, test01.formatPartyInformation(5, partyVotes, candidateVotes)	true	true	

Post condition(s) for Test:
formatPartyInformation() given a party with no candidates will throw an IOException

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __

Test Date: 3/26/24

Test Case ID#: EDOFBI_3a
Test Description: Format Balloting into partyVotes for a file with correct Ballot formatting

Name(s) of Testers: Bethany Free.am

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest
formatBallotInformation()

Automated: yes_x__ no ___

Results: Pass ___ x__ Fail _____

Preconditions for Test: formatPartyInformation() already proven to work, a file that exists in the relative path given
An ExtractDataCPL Object, partyVotes is initialized to an empty ArrayList<>, candidateVotes is initialized to an empty arrayList

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new BufferedReader Object	validFile = new BufferedReader(new FileReader(new File("src/test/java/InputFiles/OPLBallotTest01.txt")))	Object created	Object Created	
3	Create a new ExtractDataOPL Object	ttest01 = new ExtractDataOPL(validFile, "OPL");	Object Created	Object Created	
4	Instantiate partyCandidates using formatPartyInformation()	partyCandidates = test01.formatPartyInformation(5, partyVotes, candidateVotes)	Object Instantiated	Object Instantiated	
5	Call formatBallotInformation	test01.formatBallotInformation(partyVotes, candidateVotes, partyCandidates);	Nothing	Nothing	

6	Created ArrayList<ArrayList<Object>> Objects	ArrayList<ArrayList<Object>> expectedPartyVotes = new ArrayList<>();
		expectedPartyVotes.add(new ArrayList<>(Arrays.asList("Pluto", 40048)));
		expectedPartyVotes.add(new ArrayList<>(Arrays.asList("Green", 59952)));
		
		ArrayList<ArrayList<Object>> expectedCandidateVotes = new ArrayList<>();
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Becky", 20105)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Jonah", 19943)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Mariah", 19943)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Radius", 20020)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Louis", 19989)));
			Object Created	Object Created	

7	Assert expectedPartyVotes equals PartyVotes	expectedPartyVotes, partyVotes	true	true	
8	Assert expectedCandidateVotes equals candidateVotes	expectedCandidateVotes, candidateVotes	true	true	

Post condition(s) for Test:
PartyVotes will have been correctly created based on ballots

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __

Test Date: 3/26/24

Test Case ID#: EDOFBI_3b

Test Description: Format Balloting into partyVotes for a file with incorrect Ballot formatting

Name(s) of Testers: Bethany Free.am

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest
formatBallotInformation()

Automated: yes x no

Results: Pass x Fail

Preconditions for Test: formatPartyInformation() already proven to work, a file that exists in the relative path given
An ExtractDataCPL Object, partyVotes is initialized to an empty ArrayList<>, candidateVotes is initialized to an empty arrayList

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new BufferedReader Object	validFile = new BufferedReader(new FileReader(new File("src\\test\\java\\InputFiles\\OPLBallotTest02.txt")))	Object created	Object Created	
3	Create a new ExtractDataOPL Object	test01 = new ExtractDataOPL(validFile, "OPL");	Object Created	Object Created	
4	Instantiate partyCandidates using formatPartyInformation()	partyCandidates = test01.formatPartyInformation(5, partyVotes, candidateVotes)	Object Instantiated	Object Instantiated	
5	Assert formatBallotInformation throws IOException	IOException.class, test01.formatBallotInformation(partyVotes, candidateVotes,partyCandidates)	True	True	

Post condition(s) for Test:
formatBallotInformation will have throw an IOException if the ballots are incorrect

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __

Test Date: 3/26/24

Test Case ID#: EDOFBI_3c
Test Description: Format Balloting into partyVotes for ballots that contain no "1"

Name(s) of Testers: Bethany Free.am

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest
formatBallotInformation()

Automated: yes_x__ no ___

Results: Pass x Fail

Preconditions for Test: formatPartyInformation() already proven to work, a file that exists in the relative path given
An ExtractDataCPL Object, partyVotes is initialized to an empty ArrayList<>, candidateVotes is initialized to an empty arrayList

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new BufferedReader Object	validFile = new BufferedReader(new FileReader(new File("src\\testing\\java\\InputFiles\\OPLBallotTest03.txt")))	Object created	Object Created	
3	Create a new ExtractDataOPL Object	test01 = new ExtractDataOPL(validFile, "OPL");	Object Created	Object Created	
4	Instantiate partyCandidates using formatPartyInformation()	partyCandidates = test01.formatPartyInformation(5, partyVotes, candidateVotes)	Object Instantiated	Object Instantiated	
5	Assert formatBallotInformation throws IOException	IOException.class, test01.formatBallotInformation(partyVotes, candidateVotes,partyCandidates)	True	True	

Post condition(s) for Test:
formatBallotInformation will have throw an IOException if the ballots are null

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __

Test Date: 3/26/24

Test Case ID#: EDOFBI_3d

Name(s) of Testers: Bethany Free.am

Test Description: Format Balloting into partyVotes for ballots that are null

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest
formatBallotInformation()

Automated: yes_x__ no ___

Results: Pass x Fail

Preconditions for Test: formatPartyInformation() already proven to work, a file that exists in the relative path given
An ExtractDataCPL Object, partyVotes is initialized to an empty ArrayList<>, candidateVotes is initialized to an empty arrayList

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create a new BufferedReader Object	validFile = new BufferedReader(new FileReader(new File("src\\testing\\java\\InputFiles\\OPLBallotTest04.txt")))	Object created	Object Created	
3	Create a new ExtractDataOPL Object	test01 = new ExtractDataOPL(validFile, "OPL");	Object Created	Object Created	
4	Instantiate partyCandidates using formatPartyInformation()	partyCandidates = test01.formatPartyInformation(5, partyVotes, candidateVotes)	Object Instantiated	Object Instantiated	

5	create ArrayList<ArrayList<Object>> Objects	expectedPartyVotes = new ArrayList<>();
		expectedPartyVotes.add(new ArrayList<>(Arrays.asList("Pluto", 40052)));
		expectedPartyVotes.add(new ArrayList<>(Arrays.asList("Green", 59948)));
		expectedCandidateVotes = new ArrayList<>();
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Becky", 20132)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Jonah", 20054)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Mariah", 19920)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Radius", 19826)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Louis", 20068)));			

5	Assert expectedPartyVotes equals partyVotes	expectedPartyVotes, partyVotes	True	True	

Post condition(s) for Test:
formatBallotInformation will have throw an IOException if the ballots are null

Project Name: Project 1: Voting System **Team# 02**

Test Stage: Unit _x_ System __

Test Date: 3/26/24

Test Case ID#: EDOEF_1

Name(s) of Testers: Bethany Freeman

Test Description: This tests whether a correct FileData Object was created from extracting information from the file

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Project1/src/test/java/ExtractDataOPLTest.java
extractFromFile()

Automated: yes_x___ no ___

Results: Pass __x___ Fail _____

Preconditions for Test: header has already been read, file exists, and there is an ExtractDataOPL Object

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Create new BufferedReader Object	validFile = new BufferedReader(new FileReader(new File("src\\test\\java\\InputFiles\\OPLInput01.txt")))	Object Created	Object Created	
3	Create a new ExtractDataOPL	test01 = new ExtractDataOPL(validFile, header);	Object Created	Object Created	
4	Create a new FileData Object	FileData test = test01.extractFromFile();	Object Created	Object Created	
5	Assert test.getElectionType() equals “OPL”	"OPL", test.getElectionType()	true	true	
6	Assert test.getNumberSeats	4, test.getNumberSeats()	true	true	
7	Assert test.getNumberBallots	120000, test.getNumberBallots()	true	true	
8	Assert test.getNumberParties()	5, test.getNumberParties()	true	true	
9	Create HashMap<String, ArrayList<String>>	HashMap<String, ArrayList<String>> expected = new HashMap<String, ArrayList<String>>(); expected.put("Pluto", new ArrayList<>(Arrays.asList(" Becky", " Mariah"))); expected.put("Green", new ArrayList<>(Arrays.asList(" Jonah", " Radius", " Louis")));	Object Created	Object Created	
10	Assert expected equals test.getPartyCandidates	expected, test.getPartyCandidates()	true	true	
11	Create ArrayList<ArrayList<Object>> Object	ArrayList<ArrayList<Object>> expectedPartyVotes = new ArrayList<>();	Object Created	Object	

		expectedPartyVotes.add(new ArrayList<>(Arrays.asList("Pluto", 48107))); expectedPartyVotes.add(new ArrayList<>(Arrays.asList("Green", 71893)));		Created	
12	Assert expectedPartyVotes equals test.getPartyVotes	expectedPartyVotes, test.getPartyVotes()	true	true	

13	Create ArrayList<ArrayList<Object>> Object	ArrayList<ArrayList<Object>> expectedCandidateVotes = new ArrayList<>();
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Becky", 23971)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Jonah", 24014)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Mariah", 24136)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Radius", 24006)));
		expectedCandidateVotes.add(new ArrayList<>(Arrays.asList(" Louis", 23873)));
			Object Created	Object Created	

14	Assert expectedCandidateVotes equals test.getCandidateVotes()	expectedCandidateVotes, test.getCandidateVotes()	true	true	

Post condition(s) for Test:

A FileData that contains the correct information from the given file

Project Name: The project #, name of your system, and the team#

Test Stage: Indicate whether it is a unit test or a system test.

Test Date: The date the test was performed.

Test Case ID#: A unique ID is required. Decide on a naming convention and use numbering. Example: Ballot_Shuffle_1

Name(s) of Testers: List the names of anyone involved in running this test case.

Test Description: Describe briefly the test objective.

Automated: Indicate if the test is completely automated or being checked manually. (If you have methods running the tests and checking results, select "yes". If you are manually checking results, indicate manual by selecting the "no.")

Results: Indicate if the test passed or failed.

Step #: You will be listing the test steps in order. This number is the step number in the process.

Test Step Description: Details of the test step.

Test Data: What the test data will be for this step. Be clear on what the input data will be. If using a specific file, be clear on the name.

Expected Result: What result are you expecting from the program component or system.

Actual Result: What result were returned based on the test.

Post condition for Test: What will be true after the test has been run? Has the state of the system changed in any way?

Notes: Comments and notes for you and your team members.