Project 2 testinglogs
Team 17

POCandidateTests

PBI #4: Incorporation of PO election to the software Testing Number: Constructor_POCandidate
Team Member(s) Responsible:Joel Alfveby
Inputs: testcandidate = new POCandidate ();
Tests:Constructor
Outputs:PO_Candidate with desired name and party
Passed or Failed: Passed
Date:29th April 2021

PBI #4: Incorporation of PO election to the software Testing Number: GetNumVotes_POCandidate
Team Member(s) Responsible:Joel Alfveby

Inputs:
testcandidate = new POCandidate ("John", 'R');
testcandidate->GetNumVotes ();

Tests:GetNumVotes
Outputs: 0
Passed or Failed: Passed
Date:29th April 2021

PBI #4: Incorporation of PO election to the software Testing Number: GetName_POCandidate
Team Member(s) Responsible:Joel Alfveby

Inputs:
testcandidate = new POCandidate ("John", 'R');
testcandidate->GetName();

Tests:GetName
Outputs: “John”
Passed or Failed: Passed
Date:29th April 2021

PBI #4: Incorporation of PO election to the software Testing Number: AddVote_POCandidate
Team Member(s) Responsible:Joel Alfveby

Inputs:
testcandidate = new POCandidate ("John", 'R');
testcandidate->AddVote ();
testcandidate->GetNumVotes ();

Tests:AddVote
Outputs: 1
Passed or Failed: Passed
Date:29th April 2021

PBI #4: Incorporation of PO election to the software Testing Number: SetNumVotes_POCandidate
Team Member(s) Responsible:Joel Alfveby

Inputs:
testcandidate = new POCandidate ("John", 'R');
testcandidate->SetNumVotes (432);
testcandidate->GetNumVotes ();

Tests:SetNumVotes
Outputs: 432
Passed or Failed: Passed
Date:29th April 2021

POTests

PBI #4: Incorporation of PO election to the software Testing Number: Constructor_PO
Team Member(s) Responsible:Joel Alfveby
Inputs: testpo = new PO (audit);
Tests: Constructor
Outputs: 0, “PO”, 0
Passed or Failed: Passed
Date:30th April 2021

PBI #4: Incorporation of PO election to the software Testing Number: TieBreaker_PO
Team Member(s) Responsible:Joel Alfveby

Inputs:
testpo = new PO (audit);
std::vector<std::string> files;
files.push_back(“po_tie.csv”);
testpo->FindWinner(files);
std::string pike = “Pike”;
std::string foster = “Foster”;
bool one = pike.compare(testpo->GetWinner().GetName()) == 0;
bool two= foster.compare(testpo->GetWinner().GetName()) == 0;

Tests: TieBreaker
Outputs: “Pike” or “Foster”
Passed or Failed: Passed
Date:30th April 2021

PBI #4: Incorporation of PO election to the software Testing Number: AddCandidate_PO
Team Member(s) Responsible:Joel Alfveby

Inputs:
testpo = new PO (audit);
POCandidate * can1 = new POCandidate(“John”, ‘R’);
POCandidate * can2 = new POCandidate(“Susan”, ‘R’);
POCandidate * can3 = new POCandidate(“Ralph”, ‘D’);
POCandidate * can4 = new POCandidate(“Guy”, ‘I’);
testpo->AddCandidate(can1);
testpo->AddCandidate(can2);
testpo->AddCandidate(can3);
testpo->AddCandidate(can4);

Tests: AddCandidate
Outputs: 4
Passed or Failed: Passed
Date:30th April 2021

PBI #4: Incorporation of PO election to the software Testing Number: FindWinner_PO
Team Member(s) Responsible:Joel Alfveby

Inputs:
testpo = new PO (audit);
std::vector<std::string> files;
files.push_back(“po_data.csv”);
testpo->FindWinner(files);
POCandidate testcand = testpo->GetWinner()

Tests: FindWinner
Outputs: “Pike”
Passed or Failed: Passed
Date:30th April 2021

MainTests

PBI #2: Take in multiple files Testing Number: SingleFileOPLV
Team Member(s) Responsible:Joel Alfveby
Inputs: int status = system(“bash testonefile.sh”);
Tests: System with one OPLV file
Outputs: status
Passed or Failed: Passed
Date:30th April 2021

PBI #2: Take in multiple files Testing Number: MultipleFilesOPLV
Team Member(s) Responsible:Joel Alfveby
Inputs: int status = system(“bash testmultifile.sh”);
Tests: System with two OPLV files
Outputs: status
Passed or Failed: Passed
Date:30th April 2021

PBI #2: Take in multiple files Testing Number: SingleFileIRV
Team Member(s) Responsible:Joel Alfveby
Inputs: int status = system(“bash testonefileirv.sh”);
Tests: System with one IRV file
Outputs: status
Passed or Failed: Passed
Date:30th April 2021

PBI #2: Take in multiple files Testing Number: MultipleFilesIRV
Team Member(s) Responsible:Joel Alfveby
Inputs: int status = system(“bash testmultifileirv.sh”);
Tests: System with three IRV files
Outputs: status
Passed or Failed: Passed
Date:30th April 2021

Invalidation Tests

PBI #3: Invalidation of ballots that don’t meet the ranking criteria Testing Number: Invalidation_IRV1
Team Member(s) Responsible:Kanoog Moua
Inputs: irv_invalid.csv
Tests: Sends the IRV election invalid ballots that don’t meet the ranking criteria
Outputs: The invalidated ballots file with the ballot id
Passed or Failed: Passed
Date:30th April 2021

PBI #3: Invalidation of ballots that don’t meet the ranking criteria Testing Number: Invalidation_IRV2
Team Member(s) Responsible:Kanoog Moua
Inputs: irv_valid.csv
Tests: Sends the IRV election valid ballots that meet the ranking criteria
Outputs: The invalidated ballots file is empty
Passed or Failed: Passed
Date:30th April 2021

Project Name: Project 1: V-Pal Team 17

Test Stage: Unit _X_ System __

Test Date: 3/13/2021

Test Case ID#: irv_candidate_1

Name(s) of Testers: Joel Alfveby

Test Description:

This unit test, tests the methods for the irv_candidate class. The unit test file is named irv_candidate_UT.cpp contained in the testing folder. It is testing the constructor without args.

The file used for testing is the irv_candidate_UT.cpp test file located in the testing directory. Methods being tested in this file: IRVCandidate(), GetName(), GetParty(), GetCandidateID(), CheckInRace(), Eliminate(), SetMyBallots(), GetMyBallots(), GetNumVotes().

Automated: yes_X__ no ___

Results: Pass __X__ Fail________

Preconditions for Test: All of the corresponding source files of V-Pal have been compiled to run tests.

Step # 1	Test Step Description	Test Data	Expected Result	Actual Result	Notes
2	(IRVCandidateTests, ConstructorNoArgs)	name, party, id	“”, ‘\0’, 0	“”, ‘\0’, 0	constructor with no args is working
3	(IRVCandidateTests, Constructor)	name, party, id	Beth, D, 1	Beth, D, 1	constructor is working
4	(IRVCandidateTests, Eliminate)	CheckInRace()	true	true	The candidate has been initialized to stillInRace = true
5	(IRVCandidateTests, Eliminate)	Eliminate()	stillInRace = false;	stillInRace = false;	The eliminate() method works
6	(IRVCandidateTests, SetMyBallots)	std::vector <int> votes	0,0,1,0,1	0,0,1,0,1	The SetMyBallots() and GetMyBallots() is working
7	(IRVCandidateTests, GetNumVotes)	std::vector <int> votes, GetNumVotes()	GetNumVotes = 2	GetNumVotes = 2	GetNumVotes() is working

Post condition(s) for Test:
The attributes and methods for the irv_candidate class are working.

Project Name: Project 1: V-Pal **Team 17**

Test Stage: Unit _X_ System __

Test Date: 3/13/2021
Name(s) of Testers: Joel Alfveby

Test Case ID#: audit_test_1
Test Description:
This test case tests the methods for the audit.cpp file.
The unit test file audit_UT.cpp that tests the methods are located in the testing folder.

The file used for testing is the audit_UT.cpp test file located in the testing directory. Methods being tested in this file:
auditPrint()

Automated: yes_X__ no ___

Results: Pass __X___ Fail________

Preconditions for Test: All of the corresponding source files of V-Pal have been compiled to run tests.

Step #	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
1					
2	First test in the unit test file was to test if auditPrint works	filename(“audittest.txt), buffer (test strings)	“Here’s a test.” “It has two lines.”	“Here’s a test.” “It has two lines.”	print function is working correctly
3					
4					

Post condition(s) for Test:
The attributes and methods for the audit class are working.

The file created will have 2 lines of string printed onto it

Project Name: Project 1: V-Pal **Team 17**

Test Stage: Unit _X_ System __

Test Date: 3/13/2021

Test Case ID#: oplv_candidate_1

Name(s) of Testers: Joel Alfveby

Test Description:
This unit test, tests the methods for the oplv_candidate class. The unit test file is named oplv_candidate_UT.cpp contained in the testing folder.
It is testing the constructor,SetCurPartyRank,SetNumVotes,AddVote,SetWonSeatand AwardSeat.

The file used for testing is the oplv_candidate_UT.cpp test file located in the testing directory. Methods being tested in this file: GetName(),GetParty(),GetCurPartyRank(),GetNumVotes(),SetCurPartyRank(),AddVote(),SetNumVotes(),GetWonSeat(),SetWonSeat(),AwardSeat()

Automated: yes__X_ no ___

Results: Pass ___X__ Fail________

Preconditions for Test: All of the corresponding source files of V-Pal have been compiled to run tests.

Step # 1	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
2	first test is to test of the constructor is working	name(“John”),party(“R”)	“John” “R”	“John” “R”	The constructor works.
3	To test if the method SetCurPartyRank is working.	name(“John”), party(“R”),partyrank(4)	“John” “R” 4	“John” “R” 4	Setter for party rank works
4	test is to test if AddVote method is working	name(“John”),party(“R”),number of votes added(1)	“John” “R” 1	“John” “R” 1	Adding a vote to the candidate works.
5	test is to test if AddVote method is working	name(“John”),party(“R”),number of votes added(1000000)	“John” “R” 1000001	“John” “R” 1000001	Adding a vote to the remaining votes the candidate works.
6	test is to test if SetNumVotes is working	name(“John”),party(“R”),number of set votes(432)	“John” “R” 423	“John” “R” 423	Setting the number of votes is working.
7	test is to test if GetWonSeat is working	name(“John”),party(“R”) wonseat(false)	“John” “R” false	“John” “R” false	Calling GetWonseat when its false is returning false, passed.
8	test is to test if SetWonSeat is working	name(“John”),party(“R”) wonseat(true(after SetWonSeats is called))	“John” “R” true	“John” “R” true	Calling GetWonseats when its false is returning true after SetWonSeat is called.
9	test is to test if AwardSeat is working with different ranks	name(“John”),party(“R”),Current party rank(4),Amount of	“John”, “R”, 4,5, seatwon:true	“John”, “R”, 4,5, seatwon:true	Having rank 4 and with 5 seats awarded the candidate should

		seats awarded(5)			have been awarded a seat. Passed
10	test is to test if AwardSeat is working with different ranks	name(“John”),party(“R”),Current party rank(5),Amount of seats awarded(5)	“John”, “R”, 5,5, seatwon:true	“John”, “R”, 5,5, seatwon:true	Having rank 5 and with 5 seats awarded the candidate should have been awarded a seat. Passed
11	test is to test if AwardSeat is working with different ranks	name(“John”),party(“R”),Current party rank(6),Amount of seats awarded(5)	“John”, “R”, 6,5, seatwon:false	“John”, “R”, 6,5, seatwon:false	Having rank 6 and with 5 seats awarded the candidate should not be awarded a seat. Passed
12	test is to test if AwardSeat is working with different ranks	name(“John”),party(“R”),Current party rank(-1),Amount of seats awarded(5)	“John”, “R”, -1,5, seatwon:true	“John”, “R”, -1,5, seatwon:true	Having rank -1 with 5 seats awarded the candidate should be awarded a seat. (Bug?)

Post condition(s) for Test:

The attributes and methods for the oplv_candidate class are working.

Project Name: Project 1: V-Pal **Team 17**

Test Stage: Unit _X_ System __ **Test Date: 3/13/2021**

Test Case ID#: oplv_party_1 **Name(s) of Testers: Joel Alfveby**

Test Description:

This unit test, tests the methods for the oplv_candidate class. The unit test file is named oplv_party_UT.cpp contained in the testing folder.. The file used for testing is the oplv_party_UT.cpp test file located in the testing directory. Methods being tested in this file: GetName(),GetSeats(),GetRemainingVotes(),GetNumCandidates(),AddCandidate(),AddVote(),UpdateRank(),

Automated: yes__X_ no ___

Results: Pass ___X__ Fail________

Preconditions for Test: All of the corresponding source files of V-Pal have been compiled to run tests.

Step # 1	**Test Step Description**	**Test Data**	**Expected Result**	**Actual Result**	**Notes**
2	Test for the Constructor	name(“Republican”), (everything else default)	“Republican”,0,0,0	“Republican”,0,0,0	name,seats,remaining votes and number of candidates are initialized correctly.
3	Test for AddCandidate	name(“Republican”),can1Name(“John”),can2Name(“Susan”),can3Name(“Ralph”),can1Party(‘R’),can2Party(‘R’),can3Party(‘R’)	Num of candidates =3	Num of candidates =3	Given 3 candidate objects and after calling AddCandidate on each object, GetNumCandidates returns 3 thus confirming that AddCandidate works.
4	Test for UpdataRank	name(“Republican”),can1Name(“John”),can2Name(“Susan”),can3Name(“Ralph”),can1Party(‘R’),can2Party(‘R’),can3Party(‘R’),	can1Rank= 3 can2Rank=1 can3Rank= 2	can1Rank= 3 can2Rank=1 can3Rank= 2	After calling AddVote to each candidate a number of times, UpdateRank ranks them perfectly in order of highest to lowest number of votes.

Post condition(s) for Test:

The attributes and methods for the oplv_party class are working.

Project Name: Project 1: V-Pal **Team 17**

Test Stage: Unit _X_ System __

Test Date: 3/13/2021

Test Case ID#: irv_ballot_1

Name(s) of Testers: Joel Alfveby and Ramanish Singh

Test Description: This tests that the methods and attributes are working correctly for the irv_ballot class in irv_ballot.cpp The test file is labeled irv_ballot_UT.cpp located in the testing directory.

The file used for testing is the irv_ballot_UT.cpp test file located in the testing directory. Methods being tested in this file: IRVBallot(), GetVoterNum(), GetValid(), GetVoteSize(), GetIndex(), Eliminate(), IncrementIndex(), and GetCurrentPref().

Automated: yes__X_ no ___

Results: Pass ___X__ Fail________

Preconditions for Test: All of the corresponding source files of V-Pal have been compiled to run tests.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	(IRVBallotTests, Constructor)	randomTransformedVote({2, 3, 4, 1}), voterNum=200,numCandidates(4)	IRVBallot object is created with voterNum=200, numCandidates=4, validity=true, priorityIndex=0	IRVBallot object is created with voterNum=200, numCandidates=4, validity=true, priorityIndex=0	This is the constructor
2	(IRVBallotTests, CheckVoteSize) Here we are checking if the intialized IRVBallot gives the correct votesize	randomTransformedVote({2, 3, 4, -1}), voterNum=200,numCandidates(4)	testballot->GetVoteSize() should return 3	testballot->GetVoteSize() returns 3	CheckVoteSize() method in ballot helps us to find if the ballot is still valid or exhausted
3	(IRVBallotTests, CheckIncrementIndex) This test checks if we are incrementing the priority index properly after we eliminate candidate to which this ballot belonged	randomTransformedVote({2, 3, 4, -1}), voterNum=200,numCandidates(4)	testballot->Incrementindex(); testballot->GetIndex() should return 1.	estballot->GetIndex() returns 1.	We assign the ballots to candidates at the start of the election. At that time, priorityIndex is 0. But if the candidate to which that ballot belonged is eliminated, we increment the priority index so that ballot can be shifted to the

					next rightful owner.
4	(IRVBallotTests, CheckMulipleIncrements) In the previous UT, we tested how incrementIndex() works. But we also need to discard the vote if there are no preferences left on the ballot	randomTransformedVote({2, 3, 4, -1}), voterNum=200,numCandidates(4) testballot->Incrementindex();testballot->Incrementindex();true);testballot->Incrementindex();	After three increments, the vote should become invalid i.e .testballot->Valid() should return false	.testballot->Valid() returns false	After multiple increments, we we run out of choices on the ballot, make the ballot invalid

Post condition(s) for Test:

After the successful run of his test, it is guaranteed that all methods (including the constructor) in IRVBallot class are working as expected.

Project Name: Project 1: V-Pal **Team 17**

Test Stage: Unit _X_ System __ **Test Date: 3/13/2021**

Test Case ID#: oplv_1 **Name(s) of Testers: Joel Alfveby**

Test Description: This tests that the methods and attributes are working correctly for the oplv class in oplv.cpp The test file is labeled oplv_UT.cpp located in the testing directory.

The file used for testing is the oplv_UT.cpp test file located in the testing directory. Methods being tested in this file: GetTotalCandidates(),GetTypeofElection(),GetTotalBallots(),AddCandidate(),GetNumCandidates(),GetParties(),FindWinner()

Automated: yes__X_ no ___

Results: Pass ___X__ Fail________

Preconditions for Test: All of the corresponding source files of V-Pal have been compiled to run tests.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Test for Constructoor	Values left as default	0,””,0	0,””,0	Values are initialized correctly as totalcandidates=0,typeofelection=””,totalballots=0
2	Test for AddCandidate	can1Name(“John”),can2Name(“Susan”),can3Name(“Ralph”),can4Name(“Guy”),can1Party(‘R’),can2Party(‘R’),can3Party(‘D’),can4Party(‘I’)	partiesSize=2,party0Candidates=2,party1Candidates=1	partiesSize=2,party0Candidates=2,party1Candidates=1	After all the candidates added to the party the number of parties are allocated correctly and number of candidates of each party is correctly displayed.
3	Test for FindWinner	filename(“oplv_data.csv”),	winner0=Pike winner1=Foster winner2=Borg	winner0=Pike winner1=Foster winner2=Borg	Given a oplv_data.csv file the Findwinner is called and are put in a OPLVCandidate object vector and when called Getname() for each of the vector index it returns the correct results.

Post condition(s) for Test:

The attributes and methods for the oplv class are working.

Project Name: Project 1: V-Pal Team 17

Test Stage: Unit _X_ System __ **Test Date: 3/13/2021**

Test Case ID#: irv_1 **Name(s) of Testers: Ramanish Singh and Joel Alfveby**

Test Description:
This test case tests the attributes and methods for the irv.cpp election class. The testing file for this test case is labeled irv_UT.cpp located in the testing directory.

The testing file for this test case is the irv_UT.cpp test file located in the testing directory. It will also use the audittest.txt file in the same directory in testing. The methods tested in this case include IRV(), GetTypeOfElection(), GetTotalCandidates(), GetTotalBallots(), find(), TransformVote(), FindWinner(), GetWinner(), and SumVector().

Automated: yes_X__ no ___

Results: Pass __X___ Fail________

Preconditions for Test: All of the corresponding source files of V-Pal have been compiled to run tests.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	(IRVTests, Constructor) Testing that the constructor for IRV() is working and retrieving the correct attributes	IRV(audittest.txt)	“”, 0, 0	“”, 0, 0	Constructor is working for the class. It is testing GetTypeOfElection(), GetTotalCandidates(), and GetTotalBallots().
2	(IRVTests, TestFind) This test checks the find method	randomTransformedVote[4] = {2, 3, 0, 1} size of array=4 find 4	-1	-1	The find() function is working properly to return the index of the element we are seeking
3	(IRVTests,	int	output transformedVote= (0,3,1,	transformedVote= (0,3,1, 2)	We are converting each ballot

	TestTransformVote) This test checks if the TransformVote function is working properly	randomTransformedVote[4] = {2, 3, 4, 1};	2)		vote data into transfomedVote. So that transformedVote[priorityIndex] returns the candidate id to which the ballot belongs
4	(IRVTests, TestFindWinner) In this test, we provide a csv file and check if it gives the winner we wanted it to give	("irv_data.csv") We are providing the file irv_data.csv	“Rosen” is the winner	“Rosen” is the winner	This also serves as a system test
5	(IRVTests, TestSumVector)	testvec={3,12,2}	17	17	We use this function to sum the myBallots attribute of the candidates to get how many votes they won
6	(IRVTests, TestTieBreaker) In this test, we provide the test a csv file, which has a tie, and the test should choose the winner at random	("irv2_data.csv")	“Rosen” or “Kleinberg”	“Rosen” or “Kleinberg”	There’s a tie between “Rosen” and “Kleinberg”, so the software chooses one at random

Post condition(s) for Test:

The attributes and methods for the irv class are working.

Project Name: Project 1: V-Pal **Team 17**

Test Stage: Unit __ System _X_ **Test Date: 3/13/2021**

Test Case ID#: main_test_1 **Name(s) of Testers: Ramanish Singh and Joel Alfveby**

Test Description:
This test case tests the attributes and methods for

The testing file for this test case are csv files containing IRV and OPLV election types. These files include irv_data.csv, oplv_data.csv, irv1_data.csv,irv2_data.csv,irv3_data.csv,testoplv.csv,testoplvcustom.csv and other1.csv

Automated: yes_X__ no ___

Results: Pass __X___ Fail________

Preconditions for Test: V-Pal has been compiled

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	irv_data.csv file is being tested in V-Pal's system	filename("irv_data.csv")	Winner: Rosen Party:D Votes:4	Winner: Rosen Party:D Votes:4	When given the example irv_data.csv the necessary output is returned. Runtime of program: 0.188377 seconds
2	oplv_data.csv	filename("oplv_data.csv")	Seats won by parties: Democrats won 2 seats Republican won 1 seat.. Winner of seats: Pike (D) Foster (D) Borg (R)	Seats won by parties: Democrats won 2 seats Republican won 1 seats. Winner of seats: Pike (D) Foster (D) Borg (R)	When given the example oplv_data.csv the necessary output is returned. Runtime of program: 0.17796 seconds
3	irv1_data.csv	filename("irv1_data.csv")	Winner: Rosen Party:D Votes:5	Winner: Rosen Party:D Votes:5	When given the example irv1_data.csv the necessary output is returned. Runtime of program: 0.149057 seconds
4	irv2_data.csv	filename("irv2_data.csv")	Winner: Rosen Party:D	Winner: Rosen Party:D	When given the example irv2_data.csv the necessary

			Votes:4	Votes:4	output is returned. Runtime of program: 0.142052 seconds
5	irv3_data.csv	filename(“irv3_data.csv”)	Winner: Kleinberg Party:R Votes:4	Winner: Kleinberg Party:R Votes:4	When given the example irv3_data.csv the necessary output is returned. Runtime of program: 0.146113 seconds
6	testoplv.csv	filename(“testoplv.csv”)	Seats won by parties: Democrats won 2 seats Republican won 1 seat.. Winner of seats: Pike (D) Foster (D) Borg (R)	Seats won by parties: Democrats won 2 seats Republican won 1 seat.. Winner of seats: Pike (D) Foster (D) Borg (R)	When given the example testoplv.csv the necessary output is returned. Runtime of program: 0.195818 seconds
7	testoplvcustom.csv	filename(“testoplvcustom.csv”)	Seats won by parties: Democrats won 1 seats Republican won 1 seat Winner of seats: Susan (R) Ralph (D) Guy (I)	Seats won by parties: Democrats won 1 seats Republican won 1 seat Winner of seats: Susan (R) Ralph (D) Guy (I)	When given the example testoplvcustom.csv the necessary output is returned Runtime of program: 0.169003 seconds.
8	other1.csv	filename(“other1.csv”)	Candidate name : ElvisPerry Candidate's party: I Number of votes they received: 55555	Candidate name : ElvisPerry Candidate's party: I Number of votes they received: 55555	When given the example other1.csv the necessary output is returned. The other1.csv file is a file with 100’000 ballots. Runtime of program: 523.473 seconds

Post condition(s) for Test:

The system integration has been completed, and the individual small modules are working well together.