5321 Homework 4
Fall 2019

Question Weighting:

Question 1-5 - 20 percent each

Submittal items, for each problem

Submit the following in the PDF file

1. Test case table snapshot (this is the .xlsx file)
2. JUnit pass indicator (green bar expanded so we can see values where possible)
3. JaCoCo statement green source line annotations (not JaCoCo summary)
4. Make sure to include the time stamp on your screen shots.

Include in ZIP file

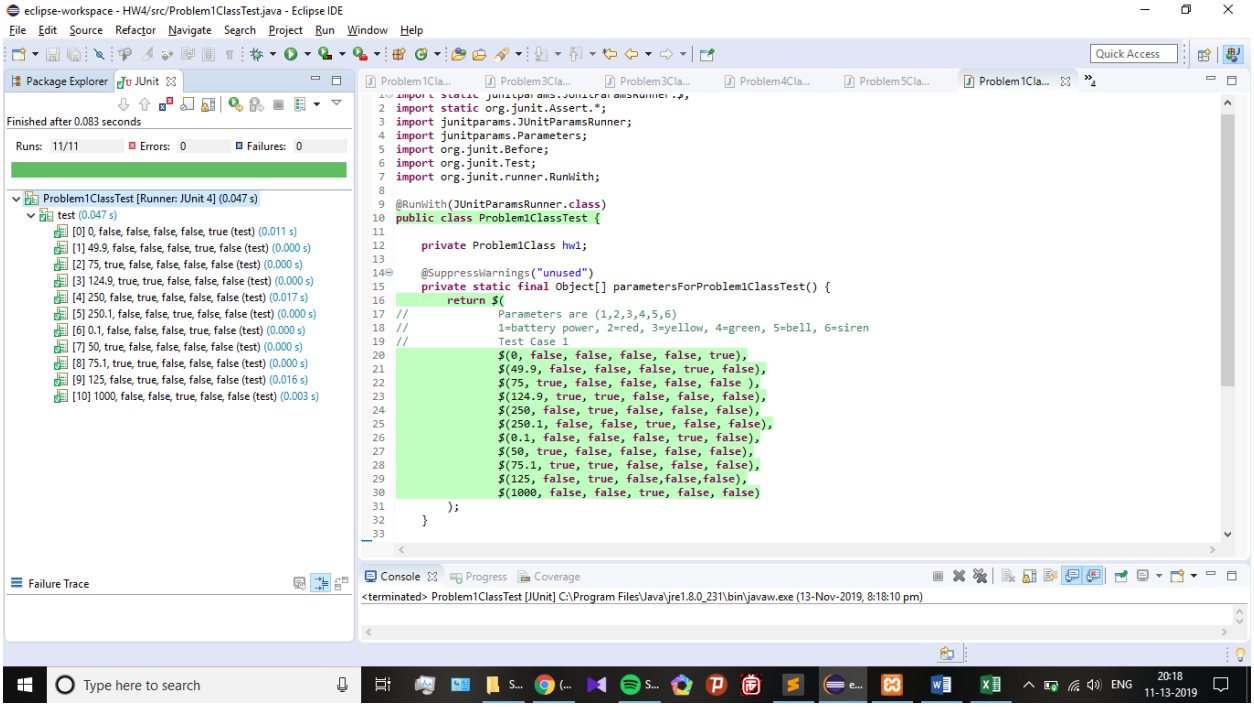
1. Your homework solution as a pdf or Word (previous item)
2. JUnit test files (make sure problem number is referenced in the file name)
3. csv files used (make sure problem number is referenced in the file name)

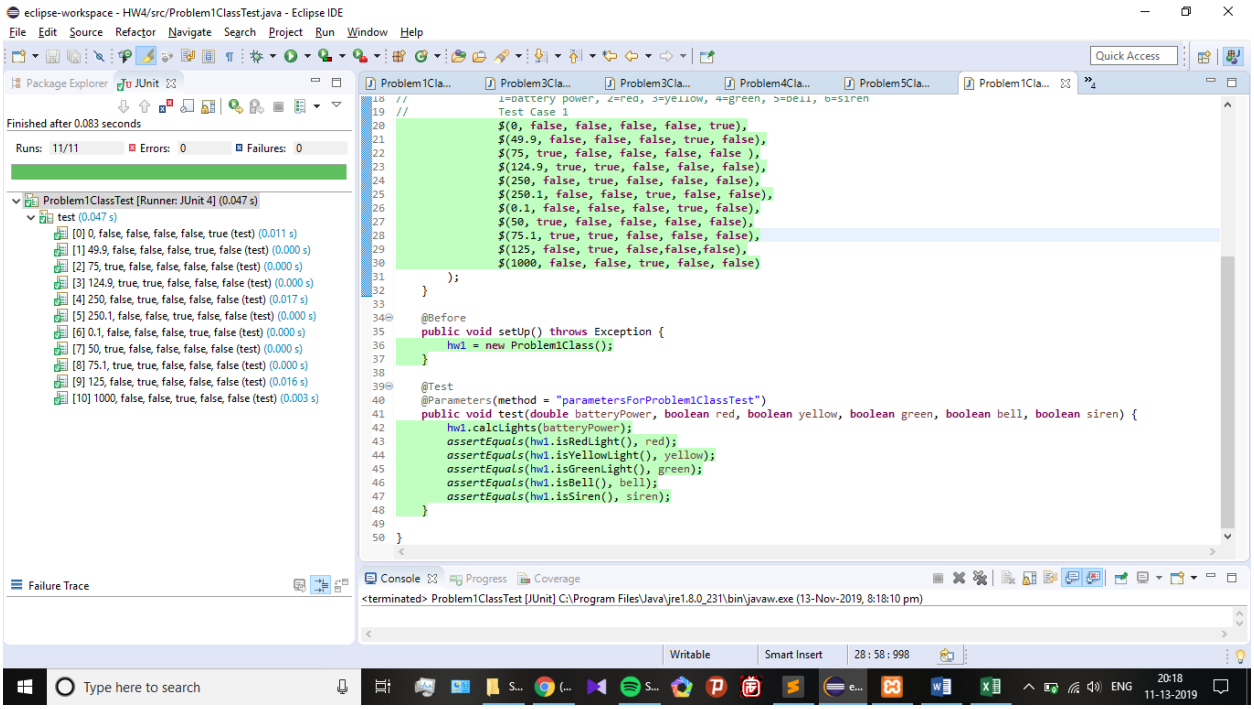
Each Java file is attached in the Blackboard zip file. Please use these files. Please do NOT modify the code - if you have to modify the code there is something wrong with your test.

Problem 1

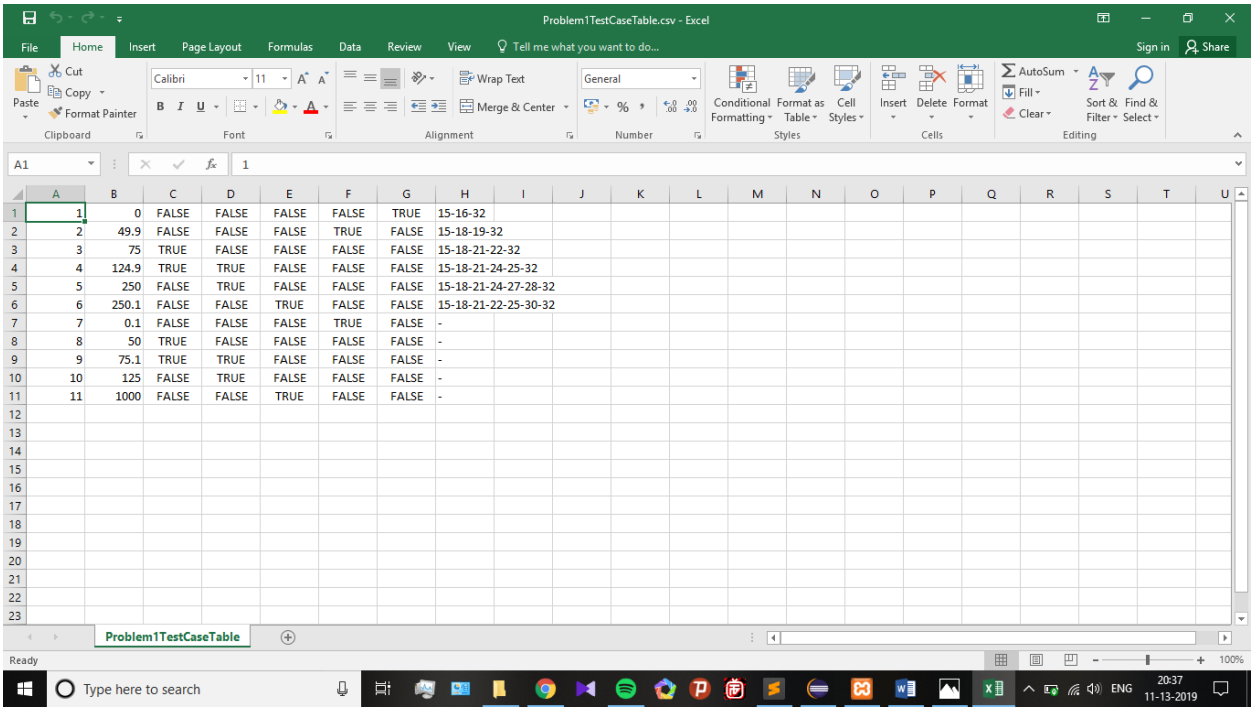
Use the code from Blackboard and the test cases we developed from Homework 3. Implement this using the **JUnitParamsRunner**. Use the posted class test cases from Homework 3.

Screenshots:





Test Case Table:

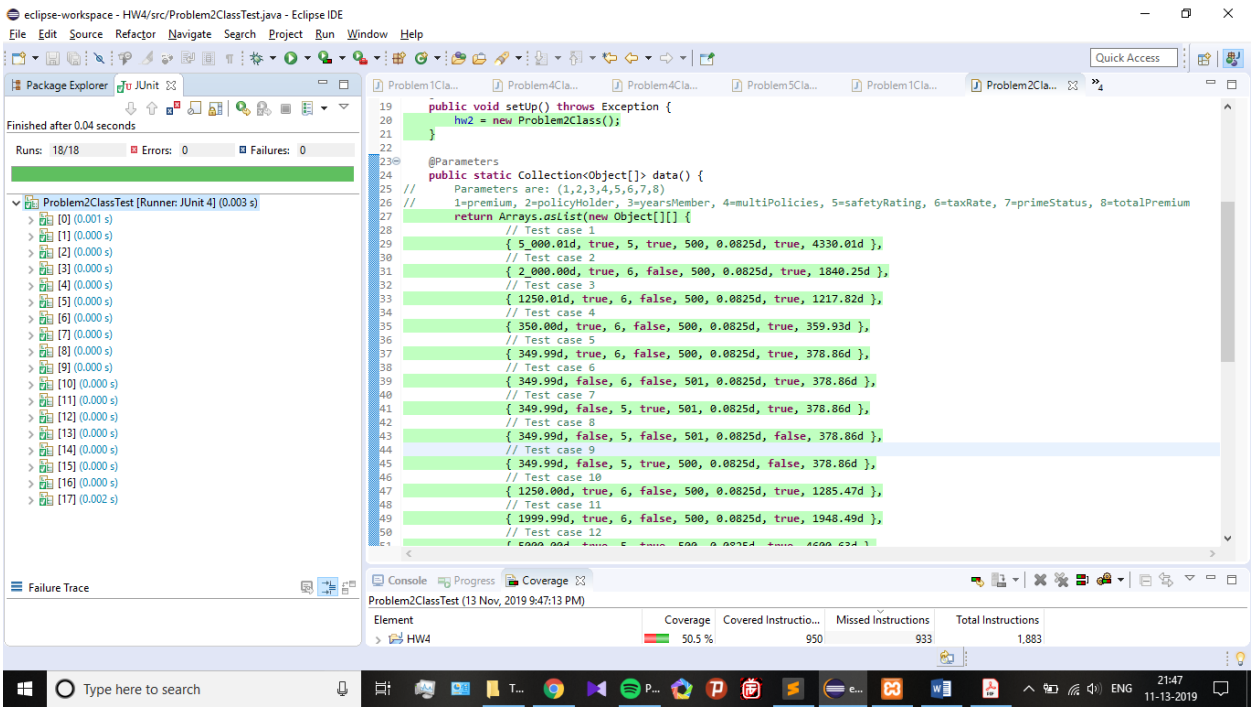


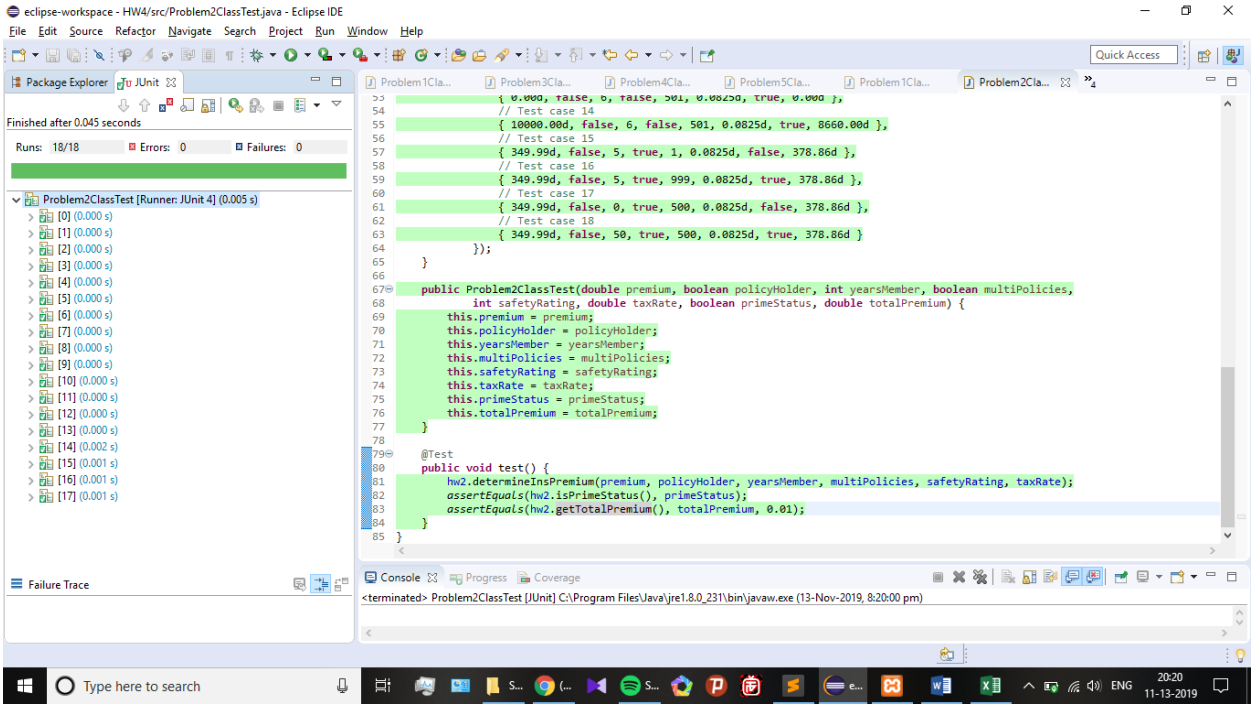
	A	B	C	D	E	F	G	H
1	1	0	FALSE	FALSE	FALSE	FALSE	TRUE	15-16-32
2	2	49.9	FALSE	FALSE	FALSE	TRUE	FALSE	15-18-19-32
3	3	75	TRUE	FALSE	FALSE	FALSE	FALSE	15-18-21-22-32
4	4	124.9	TRUE	TRUE	FALSE	FALSE	FALSE	15-18-21-24-25-32
5	5	250	FALSE	TRUE	FALSE	FALSE	FALSE	15-18-21-24-27-28-32
6	6	250.1	FALSE	FALSE	TRUE	FALSE	FALSE	15-18-21-22-25-30-32
7	7	0.1	FALSE	FALSE	FALSE	TRUE	FALSE	-
8	8	50	TRUE	FALSE	FALSE	FALSE	FALSE	-
9	9	75.1	TRUE	TRUE	FALSE	FALSE	FALSE	-
10	10	125	FALSE	TRUE	FALSE	FALSE	FALSE	-
11	11	1000	FALSE	FALSE	TRUE	FALSE	FALSE	-

Problem 2

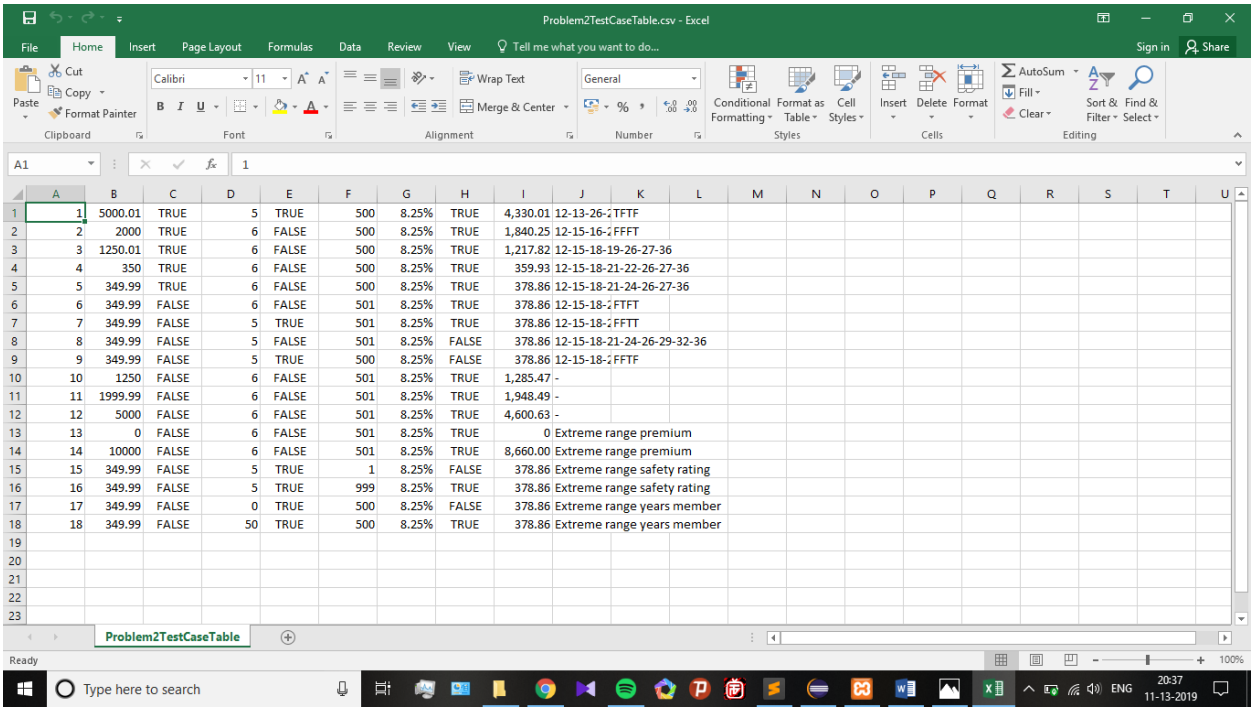
Use the code from Blackboard and the test cases we developed from Homework 3. Implement this as a **Parameterized.class** test. Use the posted class test cases from Homework 3. For the expected value of TotalPremium use a comparison threshold of **0.01** for assertEquals.

Screenshots:





Test Case Table:

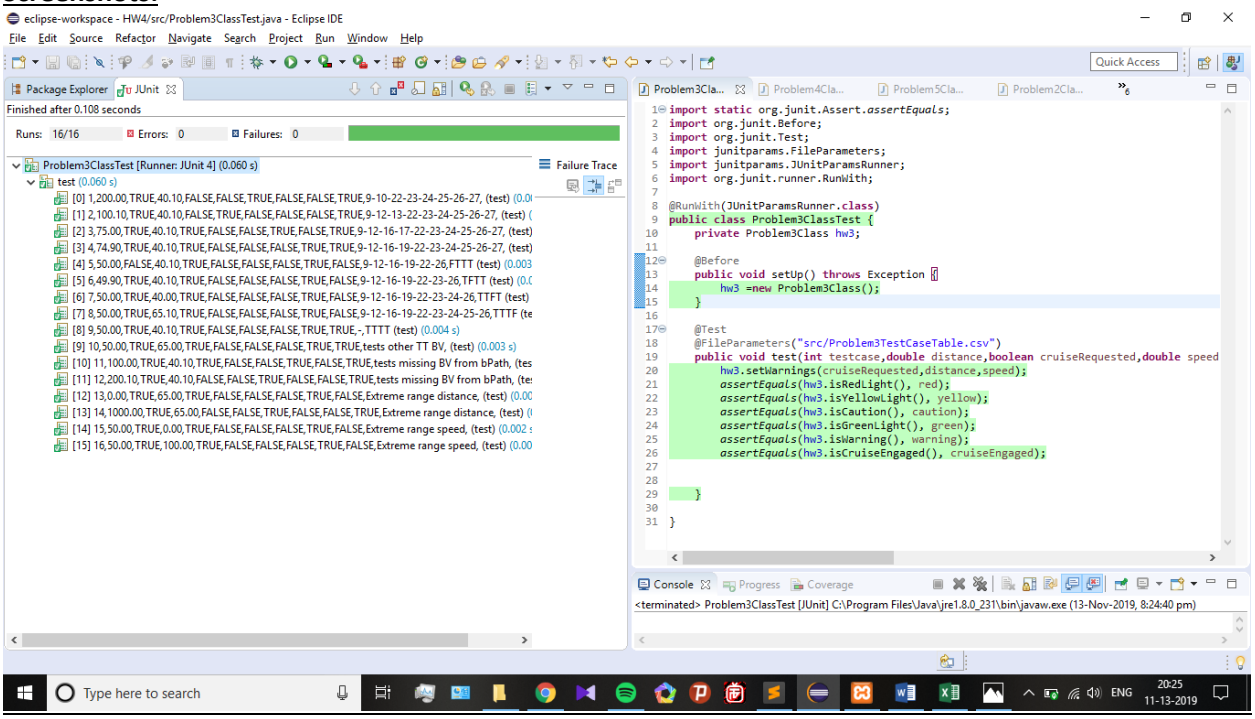


	A	B	C	D	E	F	G	H	I	J	K	L
1	1	5000.01	TRUE	5	TRUE	500	8.25%	TRUE	4,330.01	12-13-26-,	TFTF	
2	2	2000	TRUE	6	FALSE	500	8.25%	TRUE	1,840.25	12-15-16-,	FFFT	
3	3	1250.01	TRUE	6	FALSE	500	8.25%	TRUE	1,217.82	12-15-18-19-26-27-36		
4	4	350	TRUE	6	FALSE	500	8.25%	TRUE	359.93	12-15-18-21-22-26-27-36		
5	5	349.99	TRUE	6	FALSE	500	8.25%	TRUE	378.86	12-15-18-21-24-26-27-36		
6	6	349.99	FALSE	6	FALSE	501	8.25%	TRUE	378.86	12-15-18-,	FTFT	
7	7	349.99	FALSE	5	TRUE	501	8.25%	TRUE	378.86	12-15-18-,	FFFT	
8	8	349.99	FALSE	5	FALSE	501	8.25%	FALSE	378.86	12-15-18-21-24-26-29-32-36		
9	9	349.99	FALSE	5	TRUE	500	8.25%	FALSE	378.86	12-15-18-,	FFTF	
10	10	1250	FALSE	6	FALSE	501	8.25%	TRUE	1,285.47	-		
11	11	1999.99	FALSE	6	FALSE	501	8.25%	TRUE	1,948.49	-		
12	12	5000	FALSE	6	FALSE	501	8.25%	TRUE	4,600.63	-		
13	13	0	FALSE	6	FALSE	501	8.25%	TRUE	0	Extreme range premium		
14	14	10000	FALSE	6	FALSE	501	8.25%	TRUE	8,660.00	Extreme range premium		
15	15	349.99	FALSE	5	TRUE	1	8.25%	FALSE	378.86	Extreme range safety rating		
16	16	349.99	FALSE	5	TRUE	999	8.25%	TRUE	378.86	Extreme range safety rating		
17	17	349.99	FALSE	0	TRUE	500	8.25%	FALSE	378.86	Extreme range years member		
18	18	349.99	FALSE	50	TRUE	500	8.25%	TRUE	378.86	Extreme range years member		

Problem 3

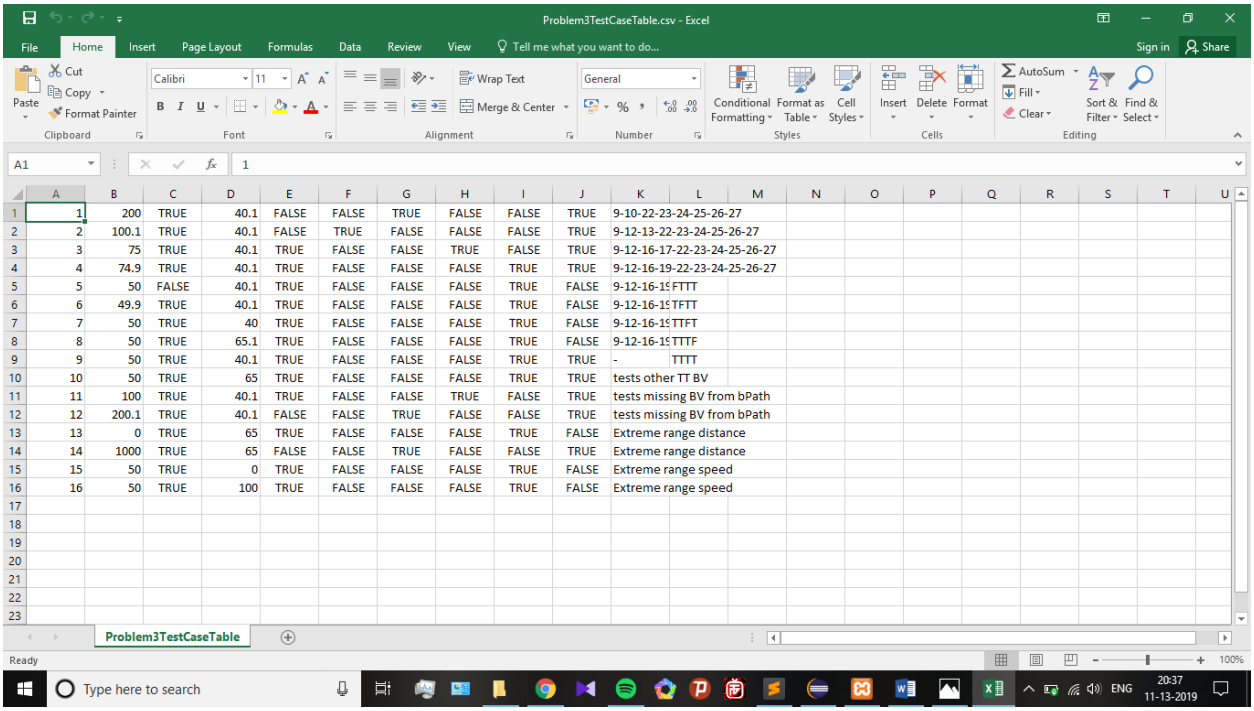
Use the code from Blackboard and the test cases we developed from Homework 3. Implement this using the **FileParameters** (read the values from a file). Use the posted class test cases from Homework 3.

Screenshots:



```
import static org.junit.Assert.assertEquals;
import org.junit.Before;
import org.junit.Test;
import junitparams.FileParameters;
import junitparams.JUnitParamsRunner;
import org.junit.runner.RunWith;

@RunWith(JUnitParamsRunner.class)
public class Problem3ClassTest {
    private Problem3Class hw3;

    @Before
    public void setUp() throws Exception {
        hw3 =new Problem3Class();
    }

    @Test
    @FileParameters("src/Problem3TestCaseTable.csv")
    public void test(int testcase,double distance,boolean cruiseRequested,double speed
        hw3.setWarnings(cruiseRequested,distance,speed);
        assertEquals(hw3.isRedLight(), red);
        assertEquals(hw3.isYellowLight(), yellow);
        assertEquals(hw3.isCaution(), caution);
        assertEquals(hw3.isGreenLight(), green);
        assertEquals(hw3.isWarning(), warning);
        assertEquals(hw3.isCruiseEngaged(), cruiseEngaged);


    }
}
```

Test Case Table:

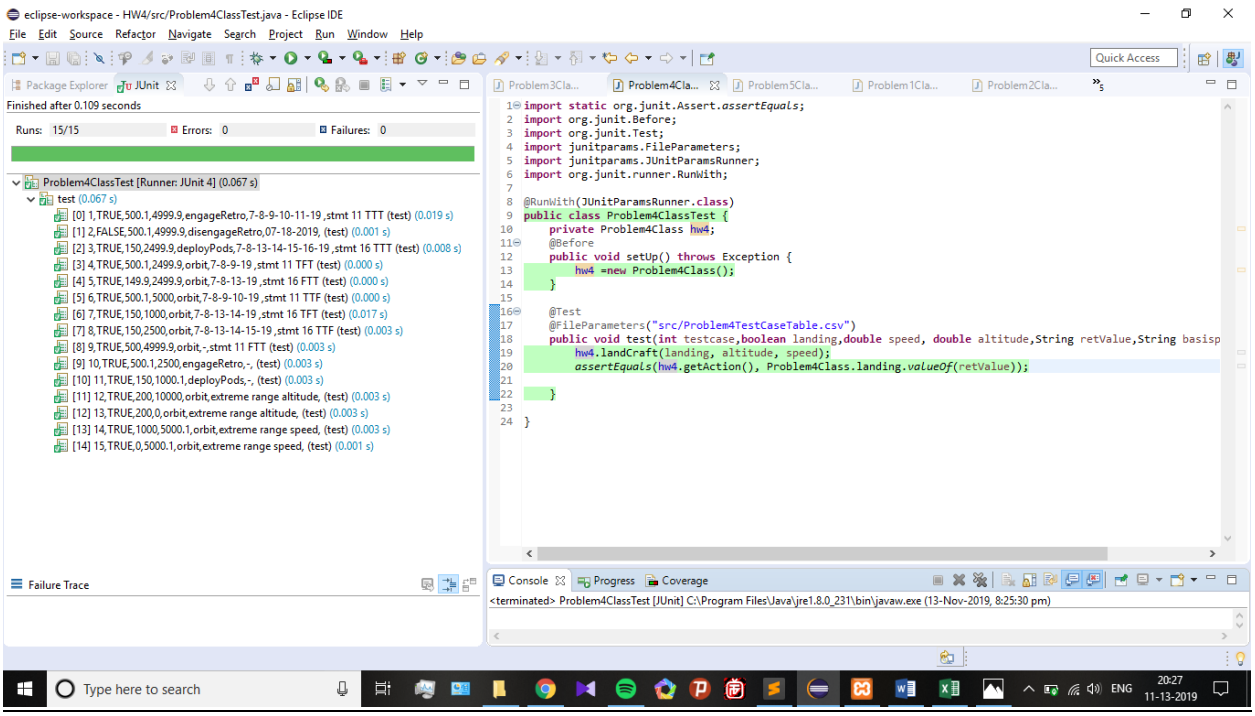


	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	200	TRUE	40.1	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	9-10-22-23-24-25-26-27		
2	2	100.1	TRUE	40.1	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	9-12-13-22-23-24-25-26-27		
3	3	75	TRUE	40.1	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	9-12-16-17-22-23-24-25-26-27		
4	4	74.9	TRUE	40.1	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	9-12-16-19-22-23-24-25-26-27		
5	5	50	FALSE	40.1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	9-12-16-1	FTTT	
6	6	49.9	TRUE	40.1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	9-12-16-1	TFTT	
7	7	50	TRUE	40	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	9-12-16-1	TTFT	
8	8	50	TRUE	65.1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	9-12-16-1	TTTF	
9	9	50	TRUE	40.1	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	-	TTTT	
10	10	50	TRUE	65	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	tests other TT BV		
11	11	100	TRUE	40.1	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	tests missing BV from bPath		
12	12	200.1	TRUE	40.1	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	tests missing BV from bPath		
13	13	0	TRUE	65	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	Extreme range distance		
14	14	1000	TRUE	65	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	Extreme range distance		
15	15	50	TRUE	0	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	Extreme range speed		
16	16	50	TRUE	100	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	Extreme range speed		

Problem 4

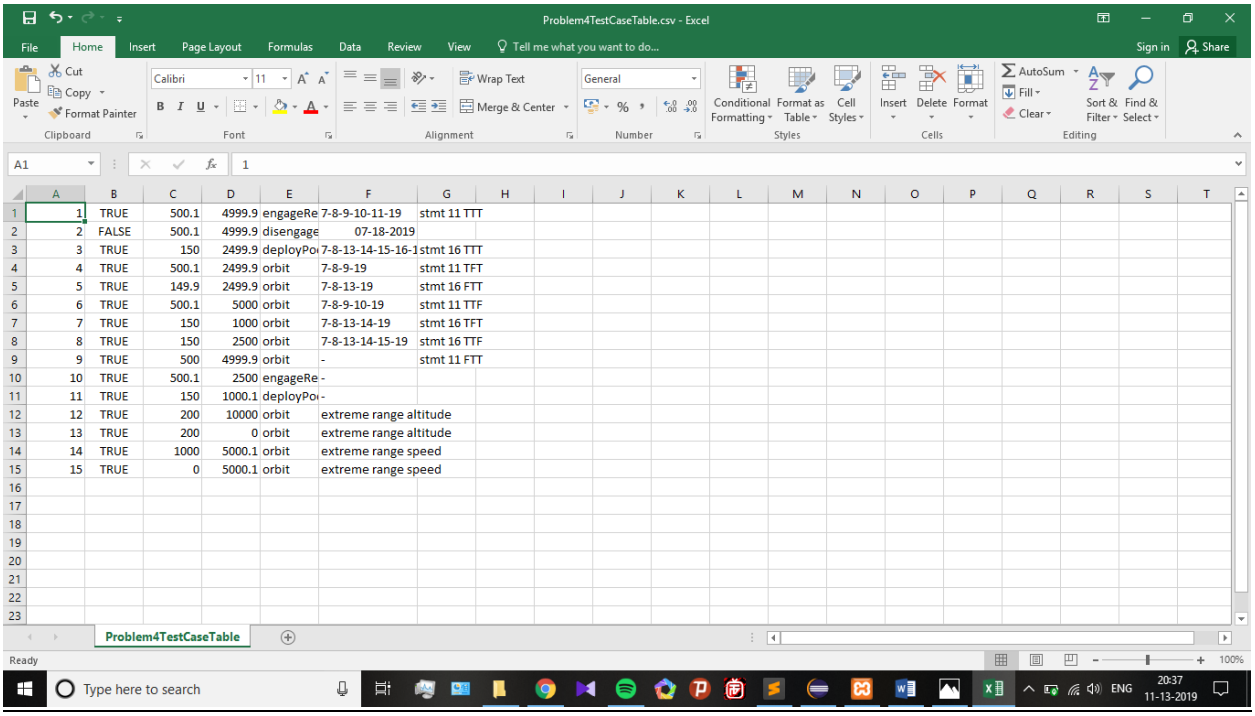
Use the code from Blackboard and the test cases we developed from Homework 3. Implement this using the **FileParameters** (read the values from a file). Use the posted class test cases from Homework.

Screenshots:




```
import static org.junit.Assert.assertEquals;
import org.junit.Before;
import org.junit.Test;
import junitparams.FileParameters;
import junitparams.JUnitParamsRunner;
import org.junit.runner.RunWith;

@RunWith(JUnitParamsRunner.class)
public class Problem4ClassTest {
    private Problem4Class hw4;
    @Before
    public void setUp() throws Exception {
        hw4 =new Problem4Class();
    }

    @Test
    @FileParameters("src/Problem4TestCaseTable.csv")
    public void test(int testcase,boolean landing,double speed, double altitude,String retValue,String basisp
        hw4.landCraft(landing, altitude, speed);
        assertEquals(hw4.getAction(), Problem4Class.landing.valueOf(retValue));

    }

}
```

Test Case Table:

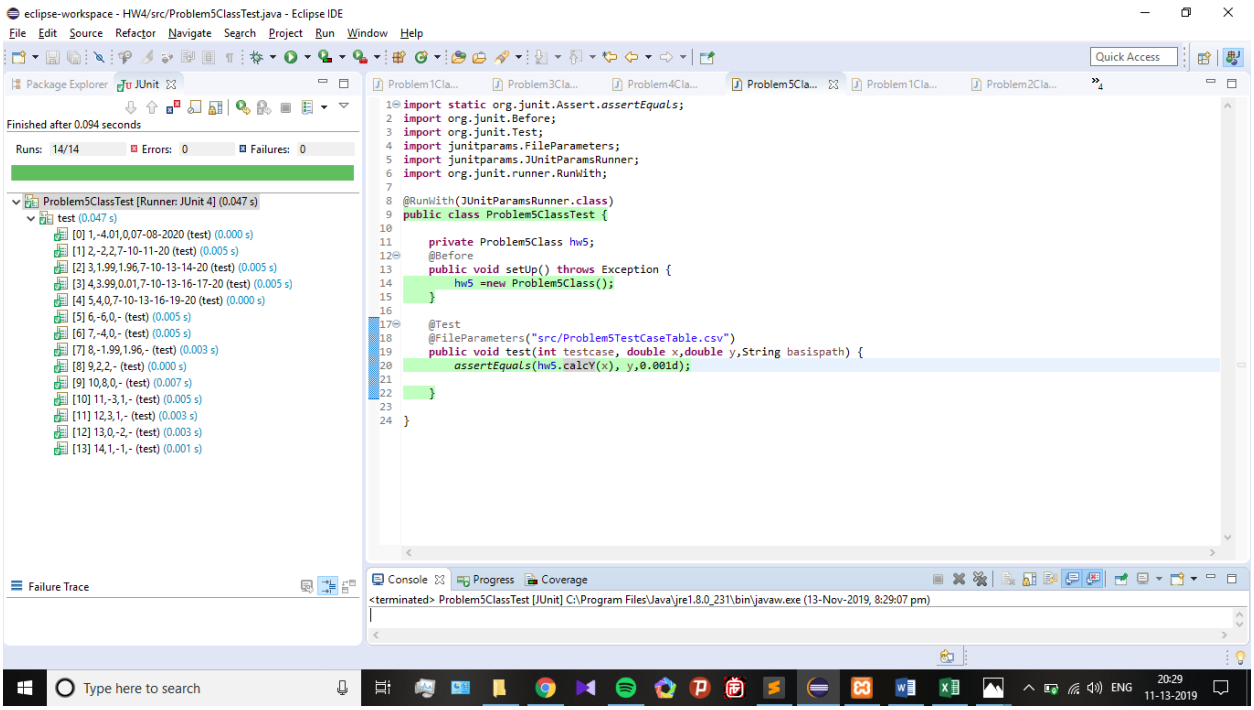


	A	B	C	D	E	F	G
1	1	TRUE	500.1	4999.9	engageRe	7-8-9-10-11-19	stmt 11 TTT
2	2	FALSE	500.1	4999.9	disengage	07-18-2019	
3	3	TRUE	150	2499.9	deployPo	7-8-13-14-15-16-1	stmt 16 TTT
4	4	TRUE	500.1	2499.9	orbit	7-8-9-19	stmt 11 TFT
5	5	TRUE	149.9	2499.9	orbit	7-8-13-19	stmt 16 FTT
6	6	TRUE	500.1	5000	orbit	7-8-9-10-19	stmt 11 TTF
7	7	TRUE	150	1000	orbit	7-8-13-14-19	stmt 16 TFT
8	8	TRUE	150	2500	orbit	7-8-13-14-15-19	stmt 16 TTF
9	9	TRUE	500	4999.9	orbit	-	stmt 11 FTT
10	10	TRUE	500.1	2500	engageRe	-	
11	11	TRUE	150	1000.1	deployPo	-	
12	12	TRUE	200	10000	orbit	extreme range altitude	
13	13	TRUE	200	0	orbit	extreme range altitude	
14	14	TRUE	1000	5000.1	orbit	extreme range speed	
15	15	TRUE	0	5000.1	orbit	extreme range speed	

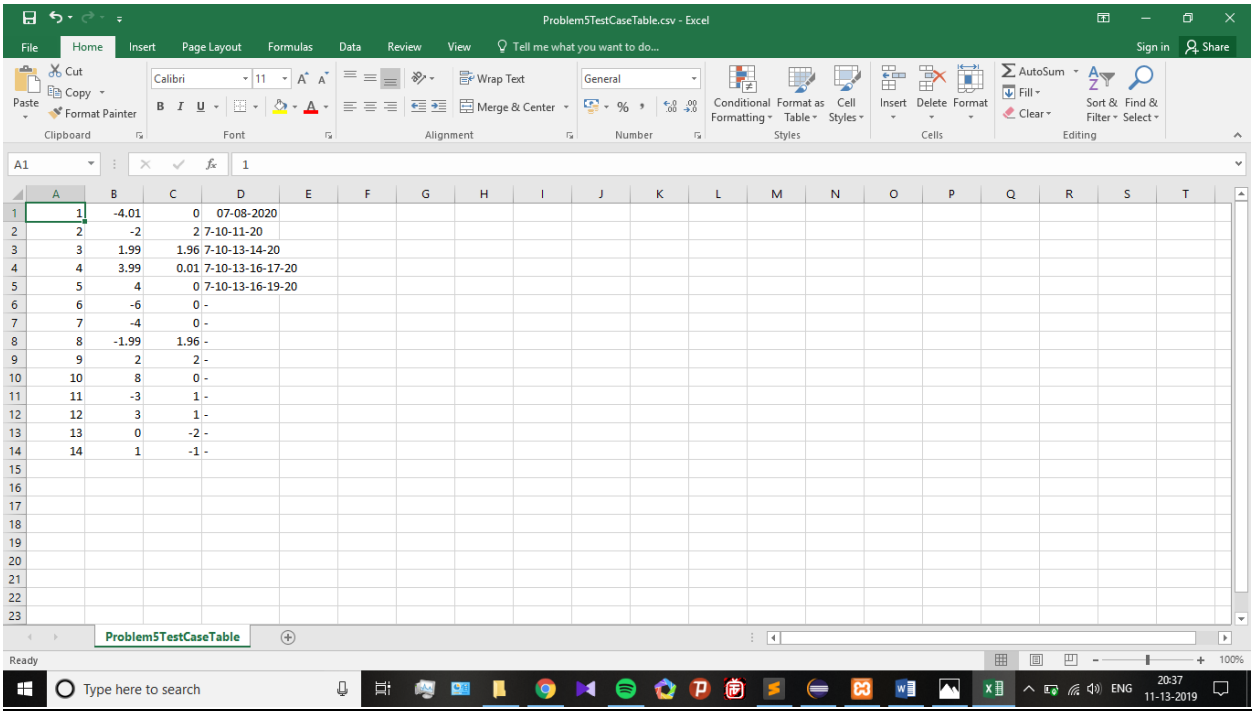
Problem 5

Use the code from Blackboard and the test cases we developed from Homework 3. Implement this using the **FileParameters** (read the values from a file). Use the posted class test cases from Homework 3. For the expected value of y use a comparison threshold of **0.001** for assertEquals.

Screenshots:



Test Case Table:



	A	B	C	D
1	1	-4.01	0	07-08-2020
2	2	-2	2	7-10-11-20
3	3	1.99	1.96	7-10-13-14-20
4	4	3.99	0.01	7-10-13-16-17-20
5	5	4	0	7-10-13-16-19-20
6	6	-6	0	-
7	7	-4	0	-
8	8	-1.99	1.96	-
9	9	2	2	-
10	10	8	0	-
11	11	-3	1	-
12	12	3	1	-
13	13	0	-2	-
14	14	1	-1	-