

CSE 5321/5321 Homework 5

Nikshith singh Varma

1001667758

1)

Test table

Test Case Number	inputs			ExpOut		Comments
	pizzaType	ordAmount	numPizzas	specialOrderStatus	total	
1	deepDish	\$500.01	4	TRUE	\$433.00	11-12-17(TTF)MCDC line 27
2	deepDish	\$250.01	4	TRUE	\$230.04	11-14-15-27
3	deepDish	\$100.01	4	TRUE	\$97.43	11-14-17-18-27
4	deepDish	\$50.01	4	TRUE	\$51.42	11-14-17-20-21-27
5	deepDish	\$50.00	5	TRUE	\$54.12	11-14-17-20-23-27(TTF)MCDC line 27
6	deepDish	\$50.00	0	TRUE	\$54.12	Extreme values for numPizzas(TTF)MCDC line 27
7	deepDish	\$500.01	1000	TRUE	\$433.00	Extreme values for numPizzas
8	!deepDish	\$50.00	4	TRUE	\$54.12	FFF MCDC line 27
9	deepDish	\$100.00	5	TRUE	\$102.83	Boundary values for ordAmount
10	deepDish	\$250.00	5	TRUE	\$243.56	Boundary values for ordAmount
11	deepDish	\$500.00	5	TRUE	\$460.06	Boundary values for ordAmount
12	deepDish	\$5,000.00	5	TRUE	\$4,330.00	Extreme values for ordAmount
13	deepDish	\$0.00	5	TRUE	\$0.00	Extreme values for ordAmount
14	deepDish	\$500.01	1000	TRUE	\$433.00	Extreme values for numPizzas

Junit and Jacoco test

The screenshot displays an IDE with the following components:

- Package Explorer:** Shows the project structure with 'Homework5' and 'Problem1ClassTest'.
- JUnit Results:** Shows a successful test run for 'Problem1ClassTest' with 13/13 runs, 0 errors, and 0 failures. The test 'test' passed in 0.077 seconds.
- Test Cases:** A list of 13 test cases with their parameters and execution times, such as '[0] 1,TRUE,500.01,4,TRUE,433.11-12-17(TTF) (test) (0.011 s)'.
- Source Code:** The 'specialOrder' method in 'Problem1Class.java' is visible, showing logic for calculating the total based on pizza type, order amount, and number of pizzas.
- Coverage:** The Jacoco coverage report for 'Problem1ClassTest' shows 35.9% coverage, with 126 covered instructions and 225 missed instructions out of 351 total instructions.

Pit Test

The screenshot shows an IDE with the following components:

- Source Code (Problem1Class.java):**

```
7 private boolean specialOrderStatus;  
8  
9 public void specialOrder (boolean deepDish, double ordAmount, int numPizzas  
10 {  
11     total=ordAmount;  
12     double taxRate=0.0825;  
13  
14     if(ordAmount>500.00)  
15         total=total*0.80*(1+taxRate);  
16     else if (ordAmount>250.00)  
17         total=total*0.85*(1+taxRate);  
18     else if (ordAmount>100.00)  
19         total=total*0.90*(1+taxRate);  
20     else if(ordAmount>50.00)  
21         total=total*0.95*(1+taxRate);  
22     else  
23         total=total*(1+taxRate);  
24  
25     specialOrderStatus = (deepDish || (ordAmount > 50.0) || (numPizzas >  
26     )  
27 }  
28  
29 public double getTotal() {  
30     return total;  
31 }
```
- PIT Mutations:**
 - 1. Removed assignment to member variable total → KILLED
 - 1. Substituted 0.0025 with 1.0 → KILLED
 - 1. changed conditional boundary → KILLED
 - 2. Substituted 500.0 with 1.0 → KILLED
 - 3. negated conditional → KILLED
 - 4. removed conditional - replaced comparison check with false → KILLED
 - 5. removed conditional - replaced comparison check with true → KILLED
 - 1. Substituted 0.8 with 1.0 → KILLED
 - 2. Substituted 1.0 with 2.0 → KILLED
 - 3. Replaced double multiplication with division → KILLED
 - 4. Replaced double addition with subtraction → KILLED
 - 5. Replaced double multiplication with division → KILLED
 - 6. Removed assignment to member variable total → KILLED
 - 1. changed conditional boundary → KILLED
 - 2. Substituted 250.0 with 1.0 → KILLED
 - 3. negated conditional → KILLED
 - 4. removed conditional - replaced comparison check with false → KILLED
 - 5. removed conditional - replaced comparison check with true → KILLED
 - 1. Substituted 0.85 with 1.0 → KILLED
 - 2. Substituted 1.0 with 2.0 → KILLED
 - 3. Replaced double multiplication with division → KILLED
 - 4. Replaced double addition with subtraction → KILLED
 - 5. Replaced double multiplication with division → KILLED
 - 6. Removed assignment to member variable total → KILLED
 - 1. changed conditional boundary → KILLED
 - 2. Substituted 100.0 with 1.0 → KILLED
 - 3. negated conditional → KILLED
 - 4. removed conditional - replaced comparison check with false → KILLED
 - 5. removed conditional - replaced comparison check with true → KILLED
 - 1. Substituted 0.9 with 1.0 → KILLED
- Coverage:**

Element	Coverage	Covered Instructio...	Missed Instruct
homework 5	35.9 %	126	

2)

Given Test table

Test Case Number	Inputs				Exp Out	Comments
	existingMember	validDiscount	memberPoints	total	expTotal	
1	FALSE	FALSE	651	2000.01	2165.01	
2	FALSE	TRUE	651	2000.00	1840.25	
3	FALSE	TRUE	649	1999.99	2164.99	
4	TRUE	FALSE	651	1999.99	2164.99	

Jacoco and Junit and pit for given test case

The screenshot displays an IDE with the following components:

- Project Explorer:** Shows a project named 'homework 5' with a package 'homework5' containing several Java files, including 'Problem2ClassTest.java'.
- Code Editor:** Displays the source code for 'Problem2Class.java'. The code defines a package 'homework5', a class 'Problem2Class', and a method 'calcTotal' that takes 'double total', 'boolean existingMember', and 'boolean validDiscount' as parameters. The method returns a double value based on a conditional expression: `return ((existingMember || validDiscount && memberPoints >= 650) && (total >= 2000.0)) ? 2165.01 : 1840.25;`
- Mutations:** A list of mutations generated by Pit, showing various changes to the code and their status (e.g., 'changed conditional boundary - SURVIVED', 'substituted 650 with 651 - SURVIVED').
- Test Results:** A summary of test results for 'Problem4ClassTest'. It shows that all 4 tests passed, with a total of 0 errors and 0 failures. The tests are listed with their parameters and execution times.
- Coverage:** A table showing the coverage of the code. The table has columns for 'Element', 'Coverage', 'Covered Instructions', 'Missed Instructions', and 'Total Instructions'. The coverage for 'homework 5' is 19.4%.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
homework 5	19.4 %	68	283	351

Derived test case table

Test Case Number	Inputs				Exp Out	
	existingMember	validDiscount	memberPoints	total	expTotal	Comments
1	FALSE	FALSE	650	\$2,000.00	\$2,165.00	
2	FALSE	TRUE	649	\$2,000.00	\$2,165.00	
3	FALSE	TRUE	650	\$2,000.00	\$1,840.25	
4	TRUE	FALSE	650	\$2,000.00	\$1,840.25	
5	TRUE	TRUE	649	\$1,999.99	\$2,164.98	
6	TRUE	TRUE	649	\$2,000.00	\$1,840.25	
7	TRUE	TRUE	651	\$2,000.00	\$1,840.25	
8	TRUE	TRUE	650	\$2,000.01	\$1,840.26	

Jacoco and JUnit test

The screenshot displays an IDE with the following components:

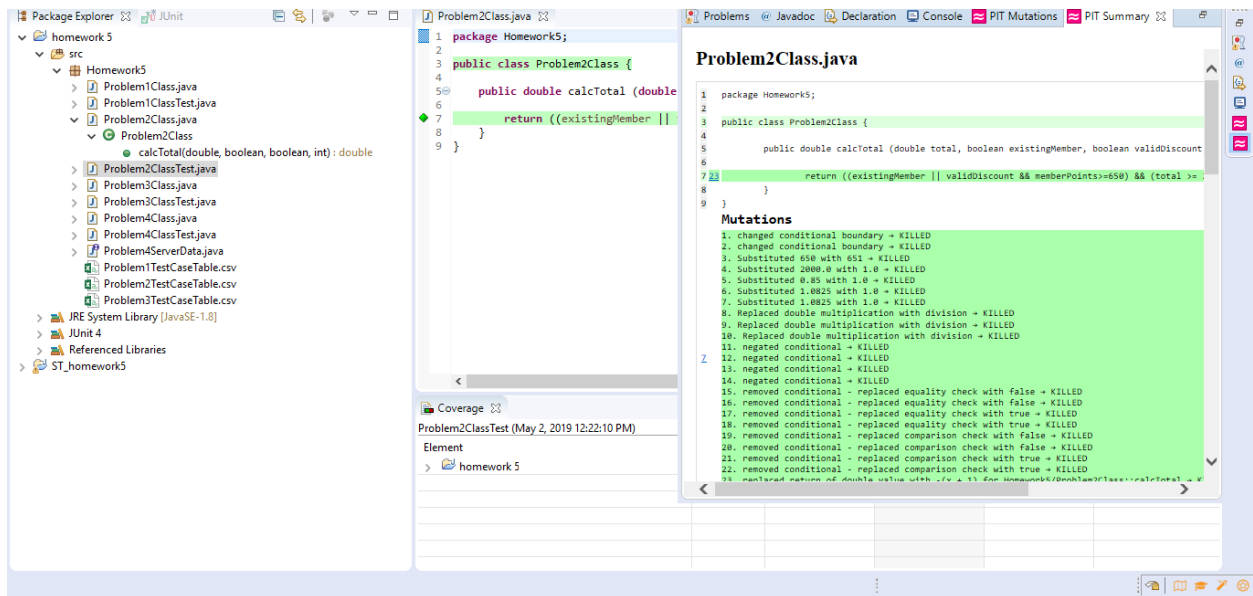
- JUnit Test Results:**
 - Finished after 0.138 seconds
 - Runs: 8/8, Errors: 0, Failures: 0
 - Test suite: Homework5.Problem2ClassTest [Runner: JUnit 4] (0.000 s)
 - Test cases (all passed):
 - [0] 1.FALSE.FALSE.650.2000.00.2165.00 (test) (0.000 s)
 - [1] 2.FALSE.TRUE.649.2000.00.2165.00 (test) (0.000 s)
 - [2] 3.FALSE.TRUE.650.2000.00.1840.25 (test) (0.000 s)
 - [3] 4.TRUE.FALSE.650.2000.00.1840.25 (test) (0.000 s)
 - [4] 5.TRUE.TRUE.649.1999.99.2164.98 (test) (0.000 s)
 - [5] 6.TRUE.TRUE.649.2000.00.1840.25 (test) (0.000 s)
 - [6] 7.TRUE.TRUE.651.2000.00.1840.25 (test) (0.000 s)
 - [7] 8.TRUE.TRUE.650.2000.01.1840.26 (test) (0.000 s)
- Code Editor:**
 - File: Problem2Class.java
 - Package: Homework5
 - Code:


```

1 package Homework5;
2
3 public class Problem2Class {
4
5     public double calcTotal (double total, boolean existingMember, boolean validDiscount, int memberPoints) {
6
7         return ((existingMember || validDiscount && memberPoints>=650) && (total >= 2_000.0)) ? 0.85*total*1.0825 :
8
9     }
          
```
- Coverage:**
 - Problem2ClassTest (May 2, 2019 12:22:10 PM)
 - Table:

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
homework 5	12.5 %	44	307	351

Pit



We got difference in Jacoco and pit in given test cases because Pit checks for boundary values and extreme values.

3)

Test Cases

Test Case Number	Input			Outputs	
	Box in Car Number	RailRoad Car Number	Shipment Number	Expected Box Number	Comment
1	1	3	5	401	8-10-14-15-1-6
2	1	3	1	391	8-14-15-16
3	1	1	1	1	8-14-16
4	1	3	10	391	TFF
5	1	3	50	401	TFT
6	1	9	5	1631	Check for loop
7	1	10	5	1821	Check for loop
8	180	10	1	2000	Extreme Values
9	1	3	100,000	401	Extreme Values

Jacoco and Junit

The screenshot shows an IDE with the following components:

- Package Explorer:** Shows the project structure with 'homework5' and 'Problem3ClassTest'.
- JUnit Results:** Shows test results for 'Problem3ClassTest' with 8 runs, 0 errors, and 0 failures. The tests passed, with execution times ranging from 0.000 s to 0.000 s.
- Code Editor:** Displays the source code for 'Problem3Class.java'. The code includes a package declaration, a class definition, and a method 'calcPrevBoxNumber' that calculates the previous box number based on input parameters.
- Coverage Table:** Shows the coverage for 'Problem3ClassTest' (May 2, 2019 12:25:55 PM). The table has columns for Element, Coverage, Covered Instructions, Missed Instructions, and Total Instructions. The coverage for 'homework5' is 29.9%.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
homework5	29.9 %	105	246	351

Pit

The screenshot shows an IDE with the following components:

- Package Explorer:** Shows the project structure with 'homework5' and 'Problem3ClassTest'.
- Code Editor:** Displays the source code for 'Problem3Class.java'.
- Coverage Table:** Shows the coverage for 'Problem3ClassTest' (May 2, 2019 12:25:55 PM). The table has columns for Element, Coverage, Covered Instructions, Missed Instructions, and Total Instructions. The coverage for 'homework5' is 29.9%.
- PIT Mutations:** Shows a list of mutations generated by PIT. The mutations are categorized into 'KILLED' and 'NOT KILLED'.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
homework5	29.9 %	105	246	351

4)

Test Case Number	Inputs				Exp Out	
	existingMember	validDiscount	memberPoints	total	expTotal	Comments
1	FALSE	FALSE	650	\$2,000.00	\$2,165.00	
2	FALSE	TRUE	649	\$2,000.00	\$2,165.00	
3	FALSE	TRUE	650	\$2,000.00	\$1,840.25	
4	TRUE	FALSE	650	\$2,000.00	\$1,840.25	
5	TRUE	TRUE	649	\$1,999.99	\$2,164.98	
6	TRUE	TRUE	649	\$2,000.00	\$1,840.25	
7	TRUE	TRUE	651	\$2,000.00	\$1,840.25	
8	TRUE	TRUE	650	\$2,000.01	\$1,840.26	

Jacoco and Jnunit

The screenshot displays an IDE with the following components:

- JUnit Test Results:**
 - Finished after 0.448 seconds
 - Runs: 8/8, Errors: 0, Failures: 0
 - Test suite: Homework5.Problem4ClassTest [Runner: JUnit 4] (0.217 s)
 - Test: test (0.217 s)
 - Test cases (all passed):
 - [0] 1, FALSE, FALSE, 650, 2000.00, 2165.00 (test) (0.101 s)
 - [1] 2, FALSE, TRUE, 649, 2000.00, 2165.00 (test) (0.021 s)
 - [2] 3, FALSE, TRUE, 650, 2000.00, 1840.25 (test) (0.023 s)
 - [3] 4, TRUE, FALSE, 650, 2000.00, 1840.25 (test) (0.038 s)
 - [4] 5, TRUE, TRUE, 1999.99, 2164.98 (test) (0.008 s)
 - [5] 6, TRUE, TRUE, 649, 2000.00, 1840.25 (test) (0.010 s)
 - [6] 7, TRUE, TRUE, 651, 2000.00, 1840.25 (test) (0.014 s)
 - [7] 8, TRUE, TRUE, 650, 2000.01, 1840.26 (test) (0.002 s)
- Code Editor:**
 - File: Problem4Class.java
 - Code snippet:

```

package Homework5;

public class Problem4Class {

    public double calcTotal (double total, boolean existingMember, boolean validDiscount, Problem4ServerData sMember
    {
        return ((existingMember || validDiscount && sMemberPoints.getMemberPoints()>=650) && (total >= 2_000.0)) ? {
    }
}

```
- Coverage:**
 - Problem4ClassTest (May 2, 2019 12:29:26 PM)
 - Element: homework 5
 - Coverage: 19.4 %
 - Covered Instructions: 68
 - Missed Instructions: 283
 - Total Instructions: 351

Pit

The screenshot displays an IDE with three main panels:

- Left Panel (Project Explorer):** Shows a project named 'homework5' with a 'src' directory. Inside 'src', there is a 'homework5' package containing several files: 'Problem1Class.java', 'Problem1ClassTest.java', 'Problem2Class.java', 'Problem2ClassTest.java', 'Problem3Class.java', 'Problem3ClassTest.java', 'Problem4Class.java', 'Problem4ClassTest.java', 'Problem4ServerData.java', and 'getMemberPoints()' (a method). There are also CSV files: 'Problem1TestCaseTable.csv', 'Problem2TestCaseTable.csv', and 'Problem3TestCaseTable.csv'. At the bottom, there are references to 'JRE System Library [JavaSE-1.8]', 'JUnit 4', and 'Referenced Libraries'.
- Middle Panel (Problem3Class.java):** Shows the source code for 'Problem3Class.java'. It defines a 'public class Problem3Class' with a 'calcTotal' method. The method takes 'double total' and 'boolean existingMember' as parameters and returns a 'double'. The code is as follows:


```

1 package Homework5;
2
3 public class Problem3Class {
4
5     public double calcTotal (double
6
7         return ((existingMember ||
8     }
9 }
      
```
- Right Panel (Problem4Class.java):** Shows the source code for 'Problem4Class.java'. It defines a 'public class Problem4Class' with a 'calcTotal' method. The method takes 'double total', 'boolean existingMember', and 'boolean validDiscount' as parameters and returns a 'double'. The code is as follows:


```

1 package Homework5;
2
3 public class Problem4Class {
4
5     public double calcTotal (double total, boolean existingMember, boolean validDiscount
6
7         return ((existingMember || validDiscount && sMemberPoints.getMemberPoints()) >
8     }
9 }
      
```
- Bottom Panel (Mutations):** A list of mutations for the 'Problem4Class.java' file. The mutations are:
 - 1. changed conditional boundary = KILLED
 - 2. changed conditional boundary = KILLED
 - 3. Substituted 550 with 651 = KILLED
 - 4. Substituted 2000.0 with 1.0 = KILLED
 - 5. Substituted 0.85 with 1.0 = KILLED
 - 6. Substituted 1.0825 with 1.0 = KILLED
 - 7. Substituted 1.0825 with 1.0 = KILLED
 - 8. Replaced double multiplication with division = KILLED
 - 9. Replaced double multiplication with division = KILLED
 - 10. Replaced double multiplication with division = KILLED
 - 11. negated conditional = KILLED
 - 12. negated conditional = KILLED
 - 13. negated conditional = KILLED
 - 14. negated conditional = KILLED
 - 15. removed call to Homework5.Problem4ServerData::getMemberPoints = KILLED
 - 16. removed conditional - replaced equality check with false = KILLED
 - 17. removed conditional - replaced equality check with false = KILLED
 - 18. removed conditional - replaced equality check with true = KILLED
 - 19. removed conditional - replaced equality check with true = KILLED
 - 20. removed conditional - replaced comparison check with false = KILLED
 - 21. removed conditional - replaced comparison check with false = KILLED
 - 22. removed conditional - replaced comparison check with true = KILLED
 - 23. removed conditional - replaced comparison check with true = KILLED