Assignment 1

Oskar Sundfors

Task 1

1. 100% code coverage:

Calendar

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
calculateDaysInMonth(int, int)		100%		100%	0	9	0	15	0	1
Calendar()		100%		n/a	0	1	0	1	0	1
Total	0 of 48	100%	0 of 15	100%	0	10	0	16	0	2

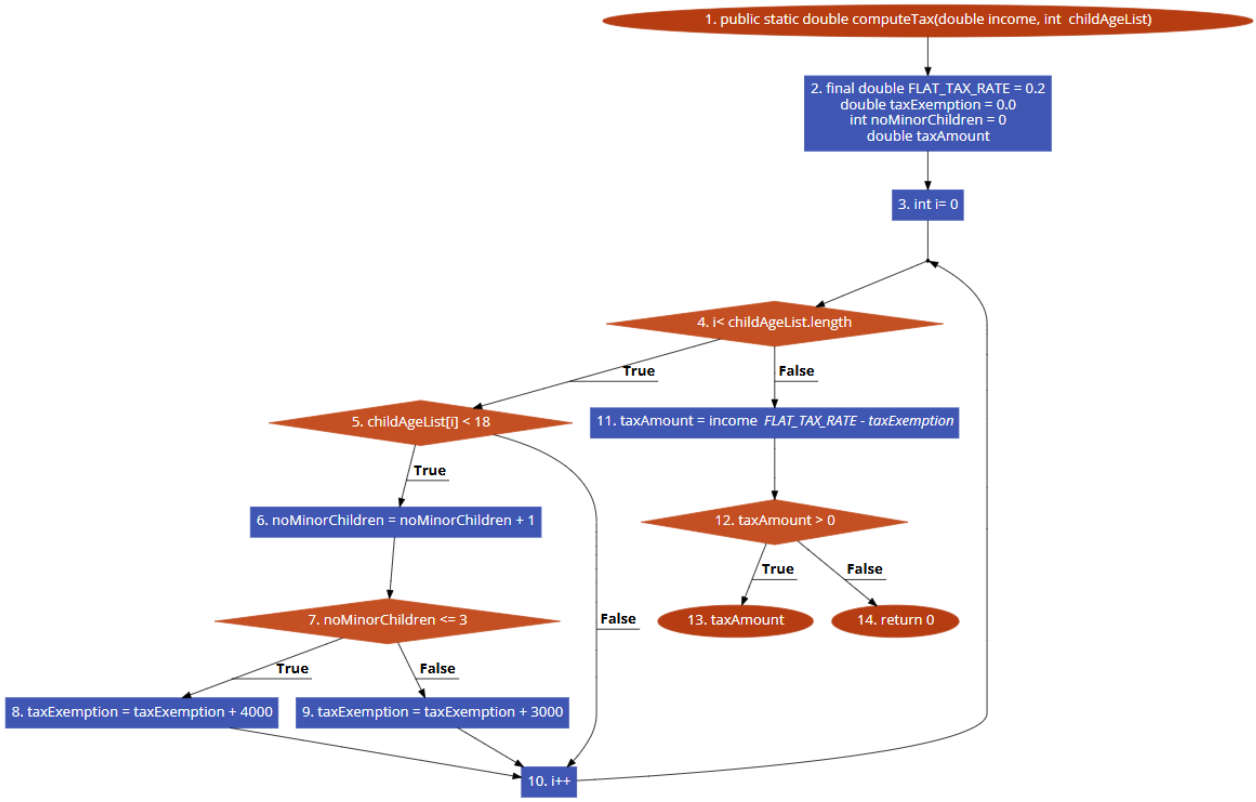
2. A minimum of 70% mutation score:

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
Calendar.java	100% 16/16	81% 13/16	81% 13/16

Task 2

1. CFG of the method:



2. Test requirements for Edge and Edge Pair coverage:

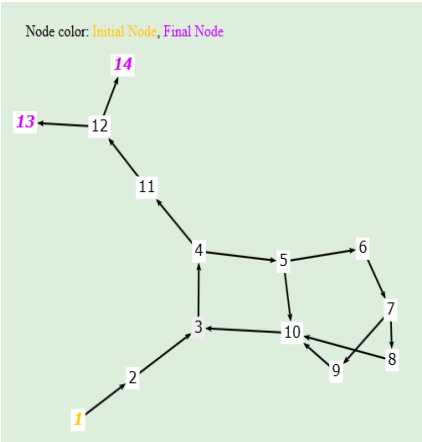
Edge test requirements:

16 requirements are needed for Edges
[1,2]
[2,3]
[3,4]
[4,11]
[11,12]
[12,13]
[12,14]
[4,5]
[5,6]
[6,7]
[7,8]
[7,9]
[8,10]
[9,10]
[5,10]
[10,3]

Edge pair test requirements:

18 requirements are needed for Edge-Pairs
1. [1,2,3]
2. [2,3,4]
3. [3,4,11]
4. [3,4,5]
5. [4,11,12]
6. [11,12,13]
7. [11,12,14]
8. [4,5,6]
9. [4,5,10]
10. [5,6,7]
11. [6,7,8]
12. [6,7,9]
13. [7,8,10]
14. [7,9,10]
15. [8,10,3]
16. [9,10,3]
17. [5,10,3]
18. [10,3,4]

Graph of the code:



3. DU-pairs for variables taxAmount and income:

DU Pairs for all variables are:

Variable	DU Pairs
taxAmount	[2,11] [2,12] [2,13]
income	[2,11]

4. Test requirements for All-Uses coverage for variable taxAmount:

All Use Coverage for all variables are:

Variable	All Use Coverage
taxAmount	[1,2,3,4,11,12,13] [1,2,3,4,11,12,13]

5. Predicates and their reachability conditions:

P	**Predicate**	**Reachability Condition**
P1	childAgeList[i] < 18	i< childAgeList.length
P2	taxAmount > 0	i >= childAgeList.length Reachable after loop on line 19
P3	noMinorChildren <= 3	i< childAgeList.length && if (childAgeList[i] < 18),
P4	i < childAgeList.length	Always reachable

6. Test requirements for Predicate Coverage for predicate on line 25:

P3 True	noMinorChildren <= 3
P3 False	noMinorChildren > 3

7 & 8. Test paths to cover all test requirements specified above:

Test ID	Test Path in graph	Input	Expected Output	EC	EPC	PC
T1	[1,2,3,4,11,12,14]	Income = 0 childAgeList = []	0		Yes	
T2	[1,2,3,4,5,6,7,8,10,3,4,5,6,7,8,10, 3,4,5,6,7,8,10,3,4,5,6,7,8,10,3,4,5, 6,7,9,10,3,4,5,10,3,4,11,12,13]	Income = 100000 childAgeList = [1,2,3,4,5,20]	taxAmount = 2000		Yes	

9. jUnit tests for tests identified above:

```java
package assigs2025.assig1;

import static org.junit.jupiter.api.Assertions.assertEquals;
import org.junit.jupiter.api.Test;

public class TestsTask2 {

    @Test
    void T1() {
        new TaxCalculator(); // line only to cover "public class TaxCalculator" line in TaxCalculator.java
        // test path: [1,2,3,4,11,12,14]
        double income = 0;
        int[] childAgeList = {};
        double expectedTax = 0;

        double result = TaxCalculator.computeTax(income, childAgeList);
        assertEquals(expectedTax, result);
    }

    @Test
    void T2() {
        // test path: [1,2,3,4,5,6,7,8,10,3,4,5,6,7,8,10,3,4,5,6,7,8,10,3,4,5,6,7,8,10,3,4,5,6,7,9,10,3,4,5,10,3,4,11,12,13]
        double income = 100000;
        int[] childAgeList = {1, 2, 3, 4, 5, 20}; // 5 minors
        double expectedTax = 2000;

        double result = TaxCalculator.computeTax(income, childAgeList);
        assertEquals(expectedTax, result);
    }
}
```

10. Code coverage and mutation score:

Code coverage:

TaxCalculator

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
computeTax(double, int[])		100%		100%	0	5	0	14	0	1
TaxCalculator()		100%		n/a	0	1	0	1	0	1
Total	0 of 52	100%	0 of 8	100%	0	6	0	15	0	2

Mutation score:

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
Calendar.java	100%	16/16	81%	13/16	81%	13/16
TaxCalculator.java	100%	15/15	80%	12/15	80%	12/15