CS265FZ Software Testing
Lab 5 – Branch Coverage

There are TWO exercises to be completed.

Two pieces of work need to be submitted:

1. Fill in this lab sheet and submit it to Moodle. You don't need to attach your source code in this form. You need to upload your source code separately.
2. Submit all the required source code to Moodle. Make sure your source code is tested in Eclipse and is executable.
3. Make sure you provide detailed comments in the source code:
    a. Identify the fault(s) in the source code.
    b. How did you fixed the fault(s)?

Program 1

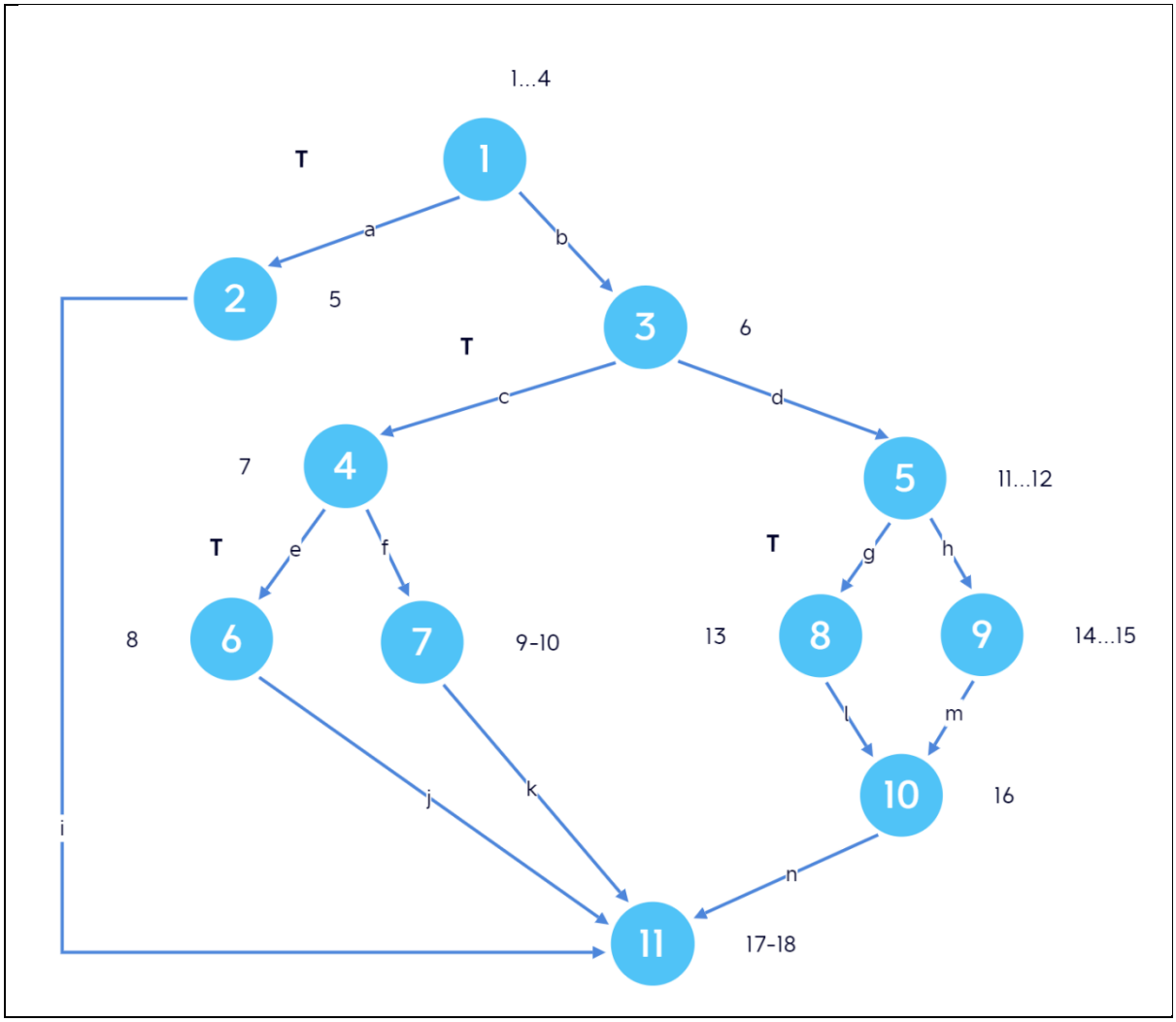
The program called ***Lab5_Program1*** has been implemented for calculating tax relief for the year 2020 for the Revenue Office in Ireland. The specification of the program is as follows: *The tax relief is applied to persons aged over 17. If a person is single, aged up to 55, the tax relief is €800, but if he/she is over 55, the tax relief is €1600. If a person is not single (married/widow/with partner, etc.), aged up to 55, the tax relief is €1600, but if he/she is over 55, the tax relief is €3200.*

Task 1

Based on the source code (*as shown in* Figure 1), construct the Control Flow Graph of the program.

```
1  public class Lab5_Program1 {
2      public int taxRelief(int age, boolean single) {
3          int taxRelief;
4          if (age < 18)
5              taxRelief = 0;
6          else if (single) {
7              if (age <= 55)
8                  taxRelief = 800;
9              else
10                 taxRelief = 1600;
11         } else {
12             if (age > 55)
13                 taxRelief = 1600;
14             else
15                 taxRelief = 3200;
16         }
17         return taxRelief;
18     }
19 }
```

Figure 1



Task 2

From the Control Flow Graph constructed in Task 1, identify the paths for a test using the Branch Coverage technique.

Test Cases	Edge
BC-1	a
BC-2	b
BC-3	c
BC-4	d
BC-5	e
BC-6	f
BC-7	g
BC-8	h
BC-9	i
BC-10	j
BC-11	k
BC-12	l
BC-13	m
BC-14	n

Task 3

Based on the paths identified in Task 2 and the program specification given at the beginning of the Problem 1, generate test data for the branch coverage test.

Test ID	**Test Cases Covered**	**Inputs**		**Expected Output**
		age	*single*	*taxRelief*
T5.1	BC-1,9	17	true	0
T5.2	BC-2,3,5,10	18	true	800
T5.3	BC-[2,3],6,11	56	true	1600
T5.4	BC-[2],4,7,12,14	56	false	1600
T5.5	BC-[2,4],8,13,14	18	false	3200

Task 4

Based on the specification given above, write your testing code in JUnit 5 to test the source code of the program provided on Moodle ("***Lab5_Program1.java***"). Make sure your test code is named as "***Lab5_Task1.java***".

Task 5

Based on the test results, provide the correct version of the "***Lab5_Program1.java***", and rename it to "***La5_Program1_Fix.java***".

Program 2

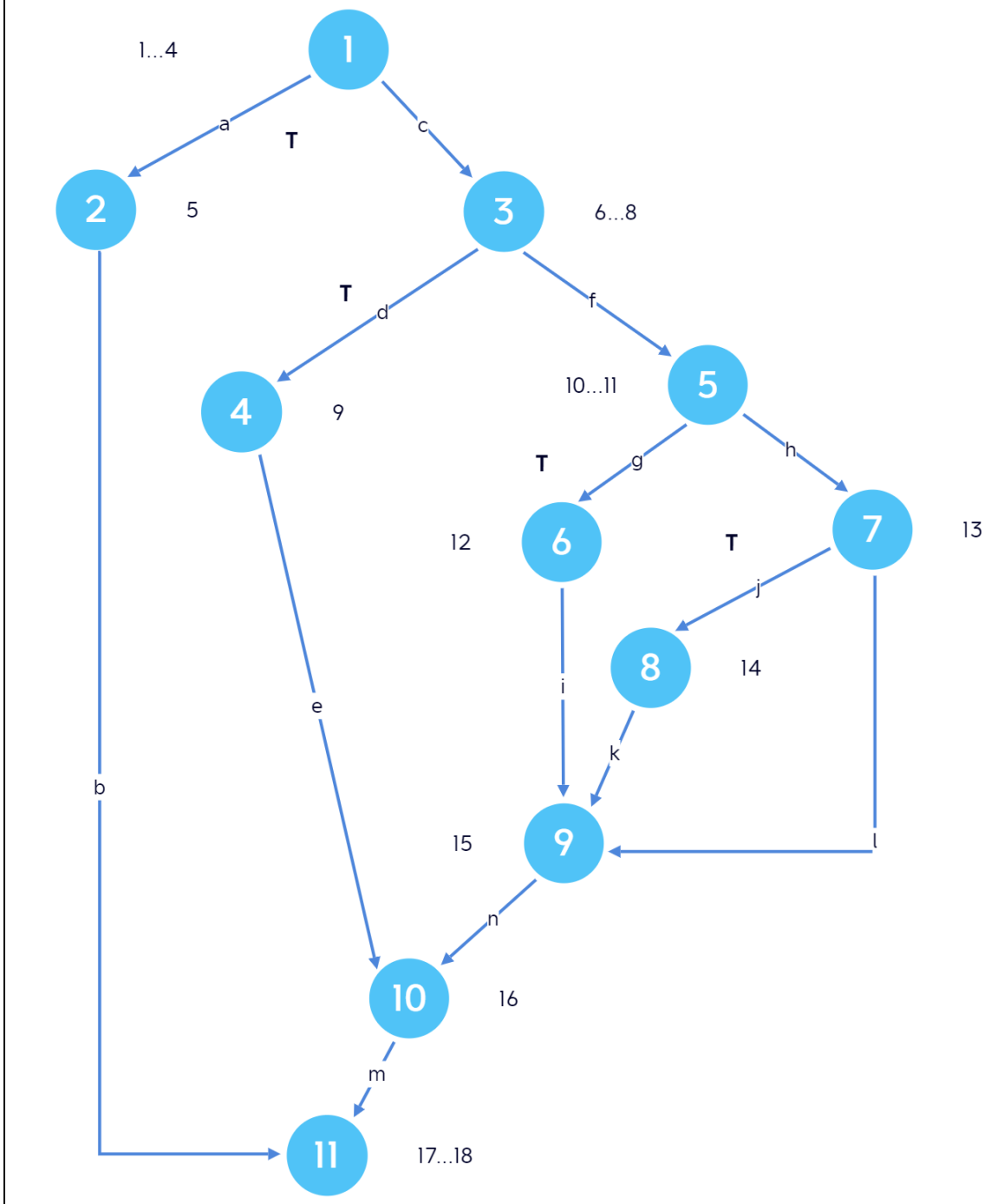
The program "***Lab5_Program2***" computes the cost of a smartphone insurance policy and outputs a value for the premium as denoted by *p*. It takes two inputs of integer *age* and Char *OS* (Operating System) type.
If the age entered is less than 16 or greater than 99 the program returns a premium of zero, *p*=0. The input for OS takes the form of 'I' for ***iOS***, 'A' for ***Android***, and 'W' for ***Windows***. If an incorrect value for the OS is entered, the program returns *p*=0.
In general, the insurance premium is €50, *p*=50. However, if a person has an iPhone and is under 25 then an extra €25 is added to the premium, *p*=75. If the person is aged between 40 and 60 (inclusive) and they have an Android phone the premium falls by €10, *p*=40. If the person is aged between 61 and 65 inclusive the premium falls by €5, *p*=45.

Task 1

Based on the source code (*as shown in* Figure 2), construct the Control Flow Graph of the program.

```
1  public class Lab5_Program2 {
2      public int phoneInsurance(int age, char OS) {
3          int p;
4          if ((age < 16) || (age > 99) || (OS != 'I' && OS != 'A' && OS != 'W'))
5              p = 0;
6          else {
7              p = 50;
8              if ((age < 25) && (OS == 'I'))
9                  p += 25;
10             else {
11                 if ((age > 40) && (age <= 60) && OS == 'A')
12                     p -= 5;
13                 else if ((age >= 61) && (age <= 65) && OS == 'W')
14                     p -= 10;
15             }
16         }
17         return p;
18     }
19 }
```

Figure 2


1...4
1
a
T
c
2
5
3
6...8
T
d
f
4
9
10...11
5
T
g
h
12
6
T
7
13
j
8
14
i
e
k
b
15
9
l
n
10
16
m
11
17...18

Task 2

From the Control Flow Graph constructed in Task 1, identify the paths for a test using the Branch Coverage technique.

Test Cases	**Edge**
BC-1	a
BC-2	b
BC-3	c
BC-4	d
BC-5	e
BC-6	f
BC-7	g
BC-8	h
BC-9	i
BC-10	j
BC-11	k
BC-12	l
BC-13	n
BC-14	m

Task 3

Based on the paths identified in Task 2 and the program specification given at the beginning of the Problem 1, generate test data for the branch coverage test.

Test ID	**Test Cases Covered**	**Inputs**		**Expected Output**
		age	*OS*	*p*
T5.1	BC-1,2	15	A	0
T5.2	BC-3,4,5,14	24	I	75
T5.3	BC-[3],6,7,9,13,[14]	63	W	45
T5.4	BC-[3,6],8,10,11,[13,14]	50	A	40
T5.5	BC-[3,6,8],12,[13,14]	50	W	50

Task 4

Based on the specification given above, write your testing code in JUnit 5 to test the source code of the program provided on Moodle ("***Lab5_Program2.java***"). Make sure your test code is named as "***Lab5_Task2.java***".

Task 5

Based on the test results, provide the correct version of the "***Lab5_Program2.java***", and rename it to "***La5_Program2_Fix.java***".