CS265FZ Software Testing
Lab 4 – Statement 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)?

In this lab, you will work on structural testing techniques and using Eclemma code coverage to see the coverage rate of your tests.

Problem 1

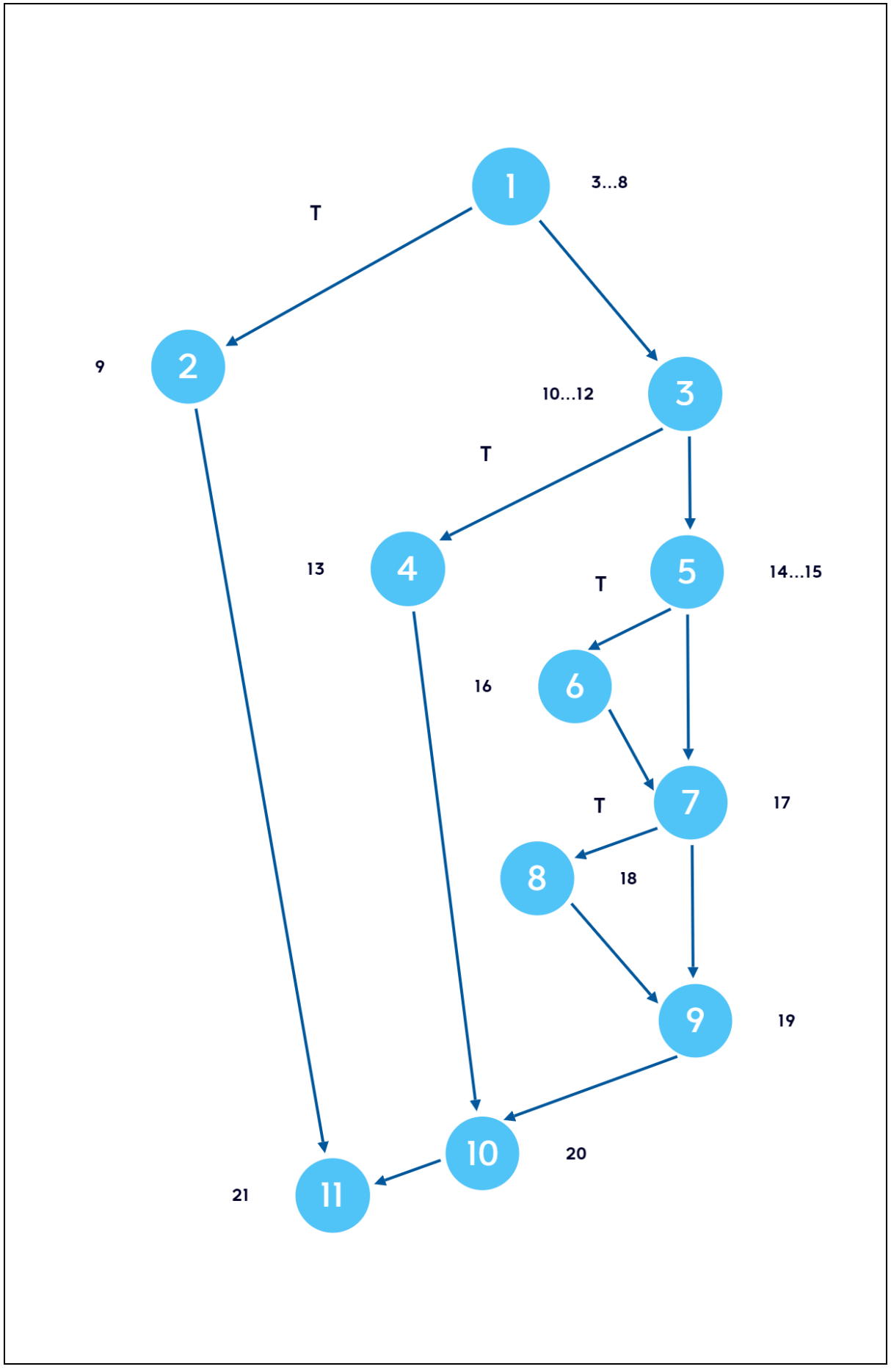
A program (Source Code: ***Lab4_Program1.java***) is used to assess the cost of a car insurance policy. It takes three inputs of age, gender, and marital status. If the age entered is less than 16 or greater than 65, the program returns a premium of zero. The input for gender takes the form of 'M' for male and 'F' for female. If an incorrect value for the gender is entered the program returns a premium of zero. In general, a premium is €500. However, if a person is male, under 25 and not married then an extra €1500 is added to the premium. If the person is female or married the premium falls by €200, and if the person is aged between 45 and 65 inclusive the premium falls by €100.

Task 1

Based on the source code (as shown in Figure 7), construct the Control Flow Graph of the program.

```
3  public class Lab4_Program1 {
4
5      public int carInsurance(int age, char gender, boolean married) {
6
7          int premium;
8          if ((age < 16) || (age > 65) || (gender != 'M' && gender != 'F'))
9              premium = 0;
10         else {
11             premium = 500;
12             if ((age < 25) && (gender == 'M') && (!married))
13                 premium += 1500;
14             else {
15                 if (married || gender == 'F')
16                     premium -= 200;
17                 if (age >= 45)
18                     premium -= 100;
19             }
20         }
21         return 0;
22     }
23 }
```

Figure 7


1
3...8
T
2
9
3
10...12
T
4
13
5
14...15
T
6
16
7
17
T
8
18
9
19
10
20
11
21

Task 2

From the Control Flow Graph constructed in Task 1, identify the paths that cover all nodes in the graph.

Test Case	Node
SC-1	1
SC-2	2
SC-3	3
SC-4	4
SC-5	5
SC-6	6
SC-7	7
SC-8	8
SC-9	9
SC-10	10
SC-11	11

Path 1: node {1, 2, 11}
Path 2: node {1, 3, 4, 10, 11}
Path 3: node {1, 3, 5, 7, 9, 10, 11}
Path 4: node {1, 3, 5, 7, 8, 9, 10, 11}
Path 5: node {1, 3, 5, 6, 7, 9, 10, 11}
Path 6: node {1, 3, 5, 6, 7, 8, 9, 10, 11}

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 statement coverage test.

Test ID	**Test Cases Covered**	**Inputs**			**Expected Output**
		age	*gender*	*married*	*premium*
T4.1	SC-1, 2, 11	8	M	false	0
T4.2	SC-[1], 3, 4, 10, [11]	20	M	false	2000
T4.3	SC-[1, 3], 5, 7, 9, [10, 11]	35	M	false	500
T4.4	SC-[1, 3, 5, 7], 8, [9, 10, 11]	50	M	false	400
T4.5	SC-[1, 3, 5], 6, [7, 9, 10, 11]	30	M	false	300
T4.6	SC-[1, 3, 5, 6, 7, 8, 9, 10, 11]	50	F	true	200

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 ("***Lab4_Program1.java***"). Make sure your test code is named as "***Lab4_Task1.java***".

Task 5

Based on the test results, provide the correct version of the "***Lab4_Program1.java***", and rename it to "***Lab4_Program1_Fix.java***".

Problem 2

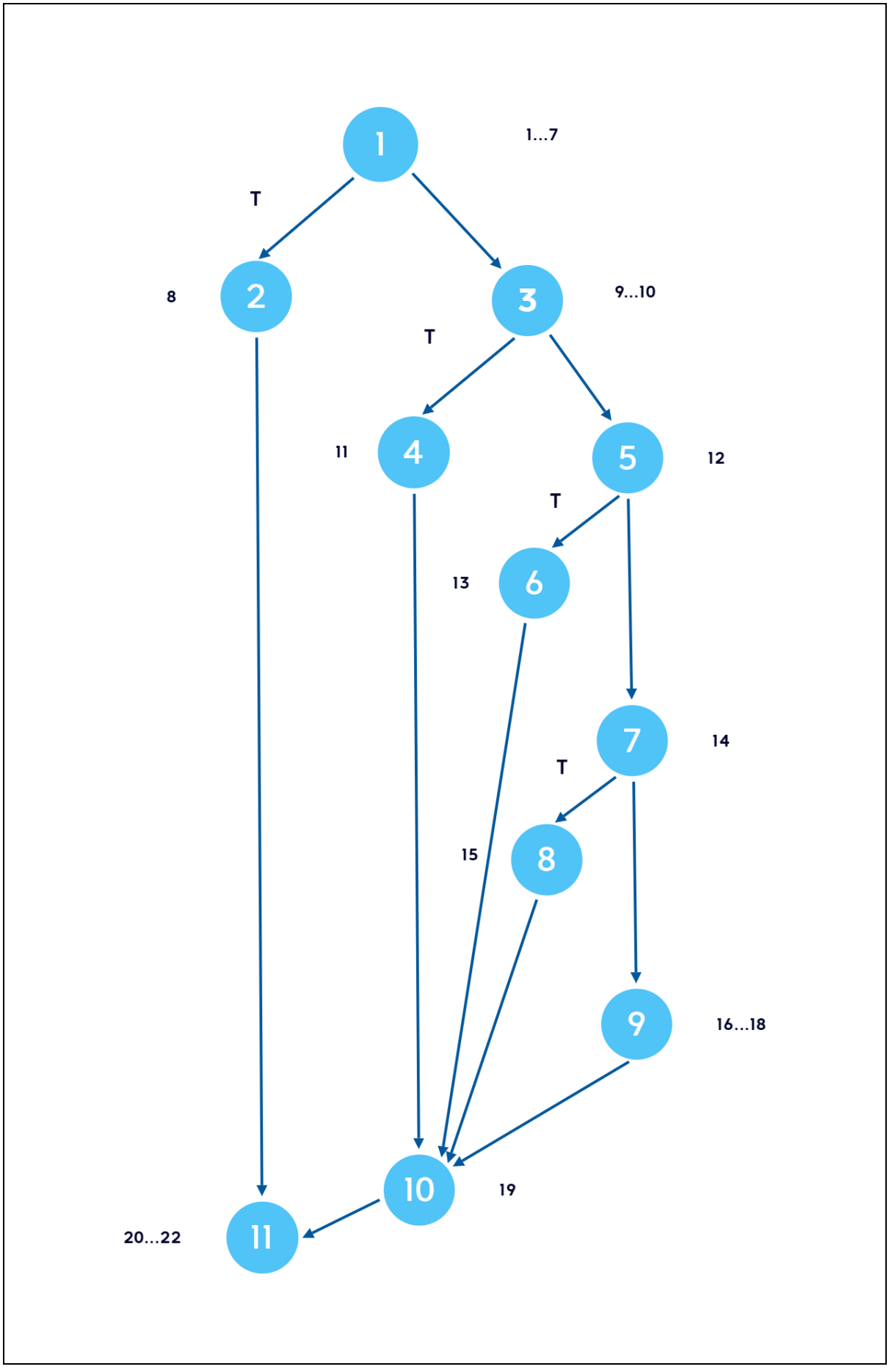
A program (**Source Code:** ***Lab4_Program2.java***) combines an exam and coursework mark into a single grade. The values for exam and coursework are integers. If the exam or coursework mark is less than 50% (< 50) then the grade returned is a "Fail". To pass the course with a "Pass, C", the student must score between 50% and 60% in the exam (50 <= exam < 60), and at least 50% in the coursework (course >= 50). They will pass the course with "Pass, B", if they score at least 60% in the exam (exam >= 60) and 50% in the coursework (course >= 50) and in addition to this, if the average of the exam and coursework is at least 70%, then they are awarded a "Pass, A". Input values that are less than 0 or greater than 100 for either the exam or coursework are invalid and the program will return a message to say "Marks out of range".

Task 1

Based on the source code (as shown in Figure 8), construct the Control Flow Graph of the program.

```
1  public class Lab4_Program2 {
2      public static String grade(int exam, int course) {
3
4          String result = "null";
5          int average;
6          average = Math.round((exam + course) / 2);
7          if ((exam < 0) || (exam > 100) && (course < 0) || (course > 100))
8              result = "Marks out of range";
9          else {
10             if ((exam < 50) || (course < 50)) {
11                 result = "Fail";
12             } else if (exam < 60) {
13                 result = "Pass,C";
14             } else if (average >= 70) {
15                 result = "Pass,A";
16             } else {
17                 result = "Pass,B";
18             }
19         }
20         return result;
21     }
22 }
```

Figure 8


1
1...7
T
2
8
3
9...10
T
4
11
5
12
T
6
13
7
14
T
8
15
9
16...18
10
19
11
20...22

Task 2

From the Control Flow Graph constructed in Task 1, identify the paths that cover all nodes in the graph.

Test Case	Node
SC-1	1
SC-2	2
SC-3	3
SC-4	4
SC-5	5
SC-6	6
SC-7	7
SC-8	8
SC-9	9
SC-10	10
SC-11	11

Path 1: node {1, 2, 11}
Path 2: node {1, 3, 4, 10, 11}
Path 3: node {1, 3, 5, 6, 10, 11}
Path 4: node {1, 3, 5, 7, 8, 10, 11}
Path 5: node {1, 3, 5, 7, 9, 10, 11}

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 statement coverage test.

Test ID	Test Cases Covered	Inputs		Expected Outputs
		exam	course	Result
T2.1	SC-1,2,4	101	101	Marks out of range
T2.2	SC-1,3,4,10,11	0	0	Fail
T2.3	SC-1,3,5,6,10,11	50	60	C
T2.4	SC-1,3,5,7,8,10,11	70	70	A
T2.5	SC-1,3,5,7,9,10,11	101	101	B

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 ("***Lab4_Program2.java***"). Make sure your test code is named as "***Lab4_Task2.java***".

Task 5

Based on the test results, provide the correct version of the "***Lab4_Program2.java***", and rename it to "***Lab4_Program2_Fix.java***".