

CS265FZ Software Testing Lab 8 – DU-Pair Coverage

There is ONE exercise 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.

Program 1

A sport centre in a university offers a range of sport activities to its staffs and students. A program “*Lab8_Program1*” calculates the annual fee as follows:

If a staff registers as a single member then the price is €450 a year, as a couple then the price is €850 a year, as a family then the price is €1000 a year. The annual fee for student is €300 and is just for only one person (single). The specification of the program is given in Table 1, and the source code of the program is given in Figure 1.

Specification

Input:

- **type (char):** 'A' (staff) or 'S' (student)
- **Scheme (integer):** 1(single), 2 (couple) , 4 (family)

Output:

- Staff single: €450
- Staff couple: €850
- Staff family: €1000
- Student: €300
- Any invalid input value will return 0.

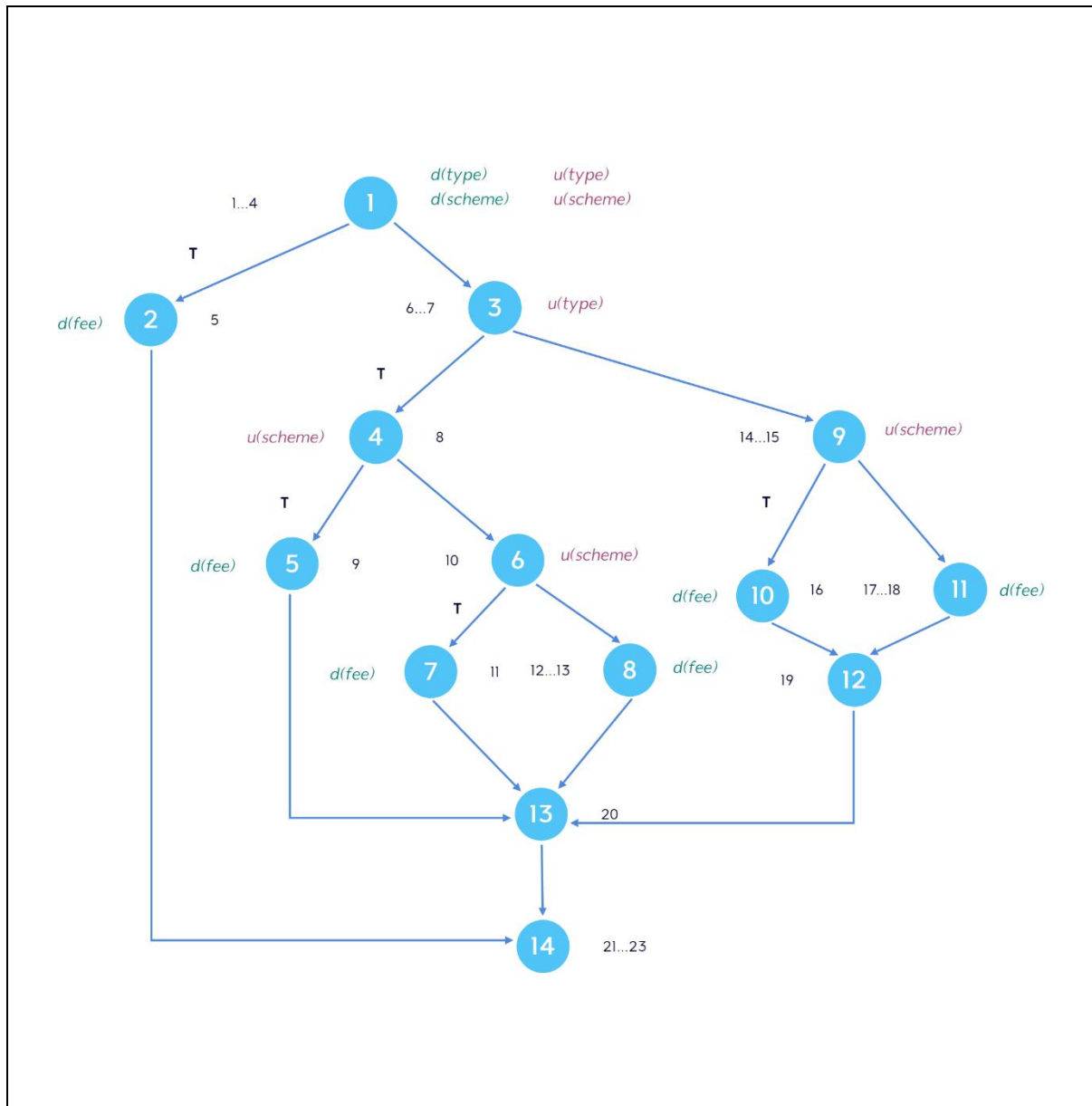
Table 1

```
1 public class Lab8_Program1 {
2     public int memberFee(char type, int scheme) {
3         int fee;
4         if ((type != 'A' && (type != 'S')) || ((scheme != 1) && (scheme != 2) && (scheme != 4)))
5             fee = 0;
6         else {
7             if (type == 'A')
8                 if (scheme == 1)
9                     fee = 450;
10                else if (scheme == 2)
11                    fee = 850;
12                else
13                    fee = 1000;
14            else {
15                if (scheme == 1)
16                    fee = 300;
17                else
18                    fee = 0;
19            }
20        }
21        return fee;
22    }
23 }
```

Figure 1

Task 1

Based on the source code (as shown in Figure 1), construct a control flow graph for the program and label all the definitions and uses of all the variables. (**Hint:** don't consider a variable declaration as a definition in DU-pair).



Task 2

Based on the source code (as shown in Figure 1) and the control flow graph you created in Task 1, construct a Definition/Use Table and a table of DU-pairs for each variable.

| Variable | Definition (line number) | Use (line number) |
|---------------|--------------------------|-------------------|
| <i>type</i> | 2 | 4,7 |
| <i>scheme</i> | 2 | 4,8,10,15 |
| <i>fee</i> | 5,9,11,13,16,18 | 21 |

| Test Case | Variable | D | U |
|-----------|---------------|----|----|
| DUP-1 | <i>type</i> | 2 | 4 |
| DUP-2 | | 2 | 7 |
| DUP-3 | <i>scheme</i> | 2 | 4 |
| DUP-4 | | 2 | 8 |
| DUP-5 | | 2 | 10 |
| DUP-6 | | 2 | 15 |
| DUP-7 | <i>fee</i> | 5 | 21 |
| DUP-8 | | 9 | 21 |
| DUP-9 | | 11 | 21 |
| DUP-10 | | 13 | 21 |
| DUP-11 | | 16 | 21 |
| DUP-12 | | 18 | 21 |

Task 3

Based on the results from Task 1 and Task 2, identify suitable test data for a DU-Pair test.

| Test ID | Test Cases Covered | Inputs | | Exp. Output | Path Taken in the CFG |
|---------|--------------------|-------------|---------------|-------------|-----------------------|
| | | <i>type</i> | <i>scheme</i> | <i>fee</i> | |
| T8.1 | DUP-1,3,7 | X | 3 | 0 | 1,2,14 |
| T8.2 | DUP-[1],2,[3],4,8 | A | 1 | 450 | 1,3,4,5,13,[14] |
| T8.3 | DUP-[1,2,3],5,9 | A | 2 | 850 | [1,3,4],6,7,13,[14] |
| T8.4 | DUP-[1,2,3,5],10 | A | 4 | 1000 | [1,3,4,6],8,13,[14] |
| T8.5 | DUP-[1,2,3,5],6,11 | S | 1 | 300 | [1,3],9,10,12,13,[14] |
| T8.6 | DUP-[1,2,3,5,6]12 | S | 2 | 0 | [1,3,9],11,12,13,[14] |

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 (“*Lab8_Program1.java*”). Make sure your test code is named as “*Lab8_Task1.java*”.