

Software Design (F28SD2): Exercises in Dynamic Analysis 1

Andrew Ireland
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh

Exercise 1

Consider the following definition:

```
int calc_handicap(int age, int wins, int lastrace, boolean firstrace)
1  { int delay = 10;
2    if ((age < 25) && (wins > 0))
3      return delay;
4    else
5      {if ((lastrace > 3) || firstrace)
6        delay = delay - 5;
7        if ((age > 35) && (age < 45))
8          delay = delay - 5;
9        return delay;}}
```

Develop structural tests for `calc_handicap` based upon the following metrics:

1. statement coverage (SC)
2. decision coverage (DC)
3. condition/decision coverage (C/DC)
4. modified condition/decision coverage (MC/DC)

For each metric you should include both test case specifications as well as the associated test data.

Exercise 2

What follows is a specification of a *Triangle Analyzer* program. Based upon the specified outputs, use equivalence partitioning and boundary value analysis to derive a set of test cases.

Triangle Analyzer Specification:

Input: Three 16-bit two's complement integers (32,768, ..., 32,767) separated by spaces provided on standard input.

Behaviour: Determine if the three integers define a valid triangle, *i.e.* where each integer denotes the size of one of the sides of a triangle.

If the input does define a triangle then classify it according to the length of its sides as *scalene* (no sides equal), *isosceles* (two sides equal), or *equilateral* (all sides equal).

Output: Display on standard output the three integers provided as input together with the classification. In the case where the input does not define a triangle then display on standard output the message NOT A TRIANGLE.

Examples: 3 4 5 Scalene
6 1 6 Isosceles
5 1 2 NOT A TRIANGLE