Term Test 2

COSC 4313 3.0 Software Engineering Testing

Section M, Winter 2006

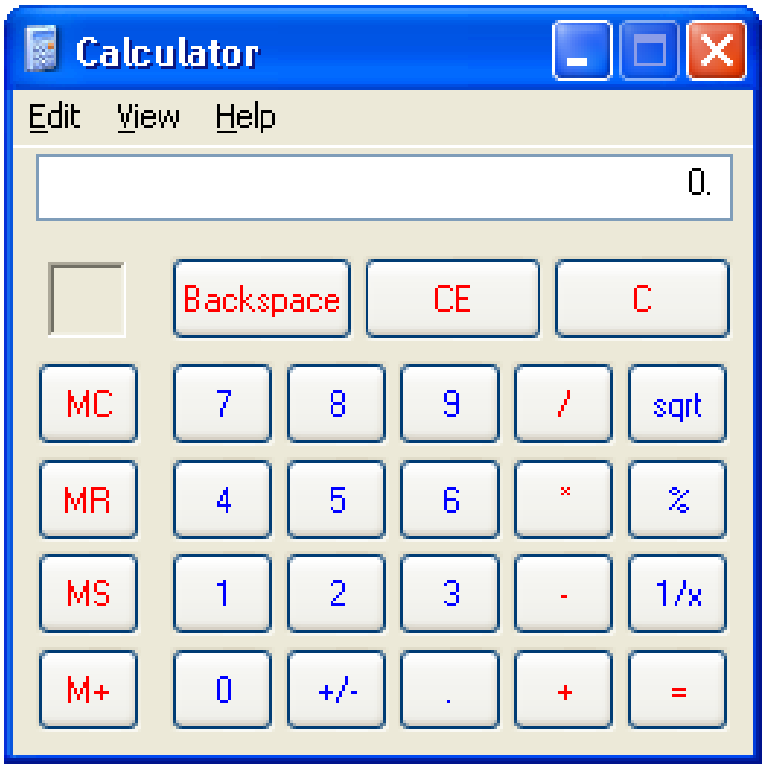
Family Name: _____

Given Name(s): _____

Student Number: |___ |___ |___ |___ |___ |___ |___ |___ |___ |

Question	Out of	Mark
Q1	20	
Q2	40	
Q3	40	
Total	100	
Letter grade		

1. **[20 marks]** Following is a screendump from a calculator application. Imagine that you have been hired to test it. Describe two GUI test cases and two functionality test cases you would run. For this question only, you do not have to derive your test cases from a testing strategy.



Any of the following test cases would be worth 5 marks:

GUI

- Buttons are disabled when appropriate, e.g. Backspace disabled when there is nothing to delete, or math operators disabled when there is no value.
- When a digit is clicked, it appears in the text area.
- Menu accelerators work correctly.
- Backspace correctly deletes one character at a time, C correctly deletes all characters.

Functionality

- Normal math operations work correctly, e.g. 2 + 2 = 4 etc.
- Overflow and underflow test cases.
- How is division by zero or square root of negative numbers handled?
- Store a number in memory, delete it from the screen, and then retrieve it from memory.

Other correct answers are also possible.

2. **[40 marks]** Consider the following method whose API indicates that it reads n numbers from the user and calculates their sum. The number n is provided by the user at the beginning.

```
1   private void calculate(){
2       int sum = 0;
3       int n = s.nextInt();
4       int i = 1;
5       while (i <= n) {
6           int number = s.nextInt();
7           sum = sum + number;
8           i = i + 1;
9       }
10      System.out.println(sum);
11  }
```

You can assume that the call to `s.nextInt()` always returns a valid integer provided by the user.

Your task is to evaluate the data flow coverage of a test suite that contains only one test case:

n = 5, number = 1,2,3,4,5

In particular, you are required to calculate the coverage of the aforementioned test suite based on the following criteria:

(a) All-Defs

(b) All-Uses

(c) All-P-Uses / Some-C-Uses

(d) All-C-Uses / Some-P-Uses

In each case, clearly descibe and explain your calculations.

For each criterion whose value is not 100%, identify a set of test cases that need to be added to the one above in order to bring the coverage value to 100%.

7 marks for correct calculation for each of the four criteria.

12 marks for determining that the extra test case to add is n=0.

We have the following definitions and uses in the program.

Variable	sum	n	i	number
Definition	2,7	3	4,8	6
C-Use	7,10	-	8	7
P-Use	-	5	5	-

All-Defs: All 6 definitions lead to at least one use, so the coverage value is 100%.

All-Uses: There are 10 D-U paths. The only path not covered by the test suite is the one from 2 to 10 for variable sum. Therefore, coverage is 90%.

All-P-Uses/Some-C-Uses: There are no P-Uses for sum and number but there is a path to a C-Use in both cases. Also, all 3 D-P-Use paths for the other two variables are covered, so the value is 100%.

All-C-Uses/Some-P-Uses: There are no C-Uses for n but there is a path to a P-Use. There are 7 D-C-use paths for the other variables, for a total of 8 paths to consider. Only the path from 2 to 10 for variable sum is missing, so coverage is 7 out of 8, or 87.5%.

3. **[40 marks]** Following is the specification for a command-line software system that computes the next date of the date entered by the user. Your task is to describe the application of any software testing strategies that you believe are appropriate in order to test this system. The derived test cases must be identified clearly in your answer. Answers that list test cases without describing how they were derived will receive a poor mark.

   The specification is provided as is. You may have to make assumptions about system behaviour that is not mentioned in the specification. If this is the case, choose reasonable ones and state them explicitly.

   Specification

   The system prompts the user for a date with the following format:

   DD/MM/YYYY

   This means that the year is always provided as 4 digits, and the month and day always as 2 digits (leading zeros are allowed). The two forward slashes must be entered as shown.

   If the format of the input is not as described above, the system outputs `Incorrect format` and exits.

   If the format is correct, but the date entered is not valid, the system outputs `Invalid Date` and exits. If the entered year is negative, or larger than 9999, the system also outputs `Invalid Date` and exits.

   In any other case, the system calculates the next date and outputs it in the same format as the input.

   Some info you probably already know

   There are 12 months in a year.
   Months 1,3,5,7,8,10,12 have 31 days.
   Months 4,6,9,11 have 30 days.
   Month 2 has 29 days in a leap year, and 28 days in a non-leap year.
   A year is leap if it is divisible by 4, unless it is a century year.
   Century years are leap only if they are divisible by 400.

10 marks: Domain analysis for input format.

4 marks: Input length. ON point 10 (2 marks), OFF points 9, 11 (2 marks).

4 marks: 3rd and 6th character. ON point /, OFF points surrounding ASCII characters.

2 marks: Other characters. ON points 0, 9, OFF points surrounding ASCII characters.

15 marks: Domain analysis for valid date.

2 marks: Year boundaries. -1 0 9999 10000

2 marks: Month boundaries. 0 1 12 13

1 mark: Day boundaries. 0 1 31 32

5 marks: All dependent boundaries for 11 months.

5 marks: Leap years. 28 and 29 Feb for years 0001, 0004, 0100, 0400 at least.

15 marks: Domain analysis for next date.

5 marks: Month boundaries. 11 months + 28 and 29 Feb in a leap year.

5 marks: Year boundaries. 31/12/1234

5 marks: Year range boundary. 31/12/9999