

Test Methodology 1DT053

Justin Pearson

10 Jan. 2013

8-13

Gimogatan 4, Sal 2

Cover Sheet

Problem no.	Solution provided	Max	Your points
1		8	
2		14	
3		10	
4		6	
5		9	
Total:		47	

Anonymous Exam Code . :

Exam Rubric

All answers to be written in English or Swedish. A mark of 50% is required for a 3, the marks 4 and 5 are evenly distributed in the interval from 50% to 100%. I will not be able to come to the exam to answer questions, if you have trouble understanding a question then please state clearly any assumptions that you have to make to answer the question.

Hjälpmedel:

Pen,pencil, ruler,rubber, dictionary, calculator.

1. General Questions on Testing.

- (a) Explain in detail the RIP (Reachability, Infection and Propagation) model and its relation to software faults and failures. You should use an example to illustrate your explanation. **(points 4)**
- (b) Explain how it is possible to have a fault in a piece of code without having a failure. **(points 2)**
- (c) Explain why the distinction between fault and failure is important when designing test cases. **(points 2)**

2. Consider the following fragment of Java code:

```
public int foobar(int x) {  
    int z=0;  
    int y=1;  
    if(x<0) { y = y*2; }  
    while(y>0 ) {  
        z = z + 1;  
        y--;  
    }  
    return(z);  
}
```

- (a) Draw a Structural Graph that represents the above piece of source code. You must indicate how each node and edge corresponds to the code. **(points 2)**.
- (b) For your graph, enumerate a complete set of test paths that guarantee node coverage. **(points 2)**.
- (c) Construct a finite set of test cases that guarantee node coverage. For each test case indicate which execution path is covered. **(points 2)**
- (d) For the above fragment of code is there any difference in requiring complete edge coverage rather than node coverage? Explain your answer. **(points 2)**
- (e) Enumerate all the prime paths of the graph you have constructed. **(points 2)**
- (f) Construct a complete set of test paths that cover all the prime paths. **(points 2)**
- (g) For each of the test paths above give a test case for the path. If the path is not feasible then say so. **(points 2)**

3. Logic Coverage. Given that the variables a, b are integers, and that P, Q are boolean variables, consider the following expression:

$$(P \ \&\& \ (a > 5)) \ || \ (\ (a < b) \ || \ (Q))$$

- (a) Define *clause coverage* (**points 2**) and for the above expression derive a set of test cases that give clause coverage. (**points 2**)
 - (b) Define what it means for a clause to determine a predicate. (**points 2**)
 - (c) There are many variations on how to define *active clause coverage*. Give two definitions and explain the differences. (**points 4**).
4. Input Space Partitioning.
- (a) For an input domain define what a partition of a domain is. (**points 2**)
 - (b) When testing the inputs to a function there are two approaches finding suitable characteristics: The interface approach, and the functionality based approach. Explain both approaches; illustrate your explanation with examples. (**points 4**)
5. General Practical Questions on Testing.
- (a) It is important to keep a good set of tests cases for regression testing, but as the project ages the regression suite can become too large. As a test engineer explain what criteria you would use to choose which test cases remain and which test cases can be taken away. (**points 3**)
 - (b) Discuss why it is often useful to have as a test engineer somebody who has not actually developed the component under test. (**points 2**)
 - (c) Describe test driven development; in particular, describe the meaning of the mantra: "Red, Green, Refactor". (**points 4**)