

**Carleton University**  
**Department of Systems and Computer Engineering**  
**SYSC-4101/5105 Midterm 2, Fall 2017 (group 1)**

Name : \_\_\_\_\_ Number : \_\_\_\_\_ Mark : \_\_\_\_ / 50

*General advices:*

- *If you need to make an assumption, you are urged to state it clearly and then proceed with your answer.*
- *If you need more space, you can use the back of the page; Simply make sure to tell the marker to consider the back page when marking.*

Answer on this examination paper.

**Question 1 [20 marks]**

Circle the correct answer.

- a) One uses abstract interpretation to identify that a piece of source code is free from divisions by zero.

Verification

Validation

- b) One conducts a usability study of the GUI of software whereby potential users are asked to use the interface and comment on it.

Verification

Validation

- c) One executes software with actual, specifically selected, input values to identify whether the software responses correspond to what users would expect.

Verification

Validation

- d) One devises tests so that each state of a state model is exercised at least once by the set of test cases, and then determines to what extent transitions of the state model are exercised.  
(Question about the all-states criterion.)

Coverage criterion

Selection criterion

- e) Given the pseudo code below, how many test cases are required to achieve 100% edge coverage?

1. Two
2. Three
3. Four
4. Five
5. More than five

```
Begin
Read(v)
If (v>85) then
  A
Else
  If (v>40) then
    B
  Endif
Endif
Endif
End
```

- f) Graph-based testing methods can only be used for source code.

- a) True
- b) False

- g) Data flow testing is a testing technique where the criteria used to design test cases
1. rely on basic path testing
  2. exercise the logical conditions in a program module
  3. select test paths based on the definitions and uses of variables
  4. focus on testing the validity of loop constructs
- h) Encapsulation of attributes and methods inside objects makes it easy to obtain object state information during testing
1. True
  2. False
- i) Graph-based testing methods can only be used on object-oriented software
1. True
  2. False
- j) Which of the following is not a control nor data flow criterion?
1. Branch coverage
  2. Prime path coverage
  3. All-Use coverage
  4. All-Choice coverage
  5. They are all control/data flow
- k) Which of the following is not a control flow criterion?
1. Edge coverage
  2. Simple round trip coverage
  3. DU path coverage
  4. Prime path coverage
  5. They are all control flow
- l) One considers a set of C functions and devises tests inputs such that each conditional branches of the code is exercised at least once, executes the tests and checks to what extent the conditions of the MC/DC criterion are satisfied. (You do not need to know the semantics of the MC/DC criterion to answer.)  
(Question about the all-branches criterion.)
- |                    |                     |
|--------------------|---------------------|
| Coverage criterion | Selection criterion |
|--------------------|---------------------|
- m) The All-Transition-Pairs criterion requires that (check the right answer)
1. Each test case exercises all the transition pairs of the graph
  2. Each test case exercises each transition pair of the graph
  3. A test suite exercises each edge of the graph multiple times
  4. A test suite exercises each transition pair of the graph at least once
  5. A test suite exercises all the edges of the graph
  6. A test suite exercises each transition pair of the graph multiple times
- n) One considers the body of a series of Java methods and devises test inputs such that each statement of the code of the class is exercised at least once and then checks to what extent exceptions have been exercised. (Question about the all-exceptions criterion.)
- |                    |                     |
|--------------------|---------------------|
| Coverage criterion | Selection criterion |
|--------------------|---------------------|

- o) One considers a set of C functions and devises tests inputs such that each conditional branches of the code is exercised at least once, executes the tests and checks to what extent the conditions of the MC/DC criterion are satisfied.  
     This is black-box testing                      This is white-box testing
- p) One devises tests so that each transition of a state model model is exercised at least once, and then executes the tests on the code.  
     This is black-box testing                      This is white-box testing
- q) When my adequate black-box test suite does not achieve 100% statement coverage
1. I know I can stop testing since my black-box tests exercise all the test requirements of the black-box criterion I selected (the test suite is adequate)
  2. I know my black-box test suite is incomplete, although adequate (for a black-box criterion), and I must reach 100% structural coverage
  3. None of the above
- r) I have to select a state-based criterion and I am only concerned about cost. According to empirical results discusses in class, I should:
1. Select the All-transition-pairs criterion
  2. Select the All-transitions criterion
  3. Select the All transition-tree criterion
  4. Select the All-state criterion
  5. None of the above, it does not matter
- s) I am using a graph-based criterion as a selection criterion; I obtain a set of 15 test paths to exercise all test requirements; When implementing my test driver to execute these test paths I discover one test path is not feasible, i.e. I cannot find test inputs to execute the path. What should I do?
1. I am happy to have one less test case to create and execute since that will reduce my testing costs
  2. I create another (feasible) test case because I need 15 test cases
  3. I do not know what to do and ask for help
  4. There must be something wrong in my graph, I try to fix it to make my test paths feasible
  5. I create another (feasible) test case to satisfy my test requirements
- t) You observe that a condition on an edge in a graph is compound. Specifically, the condition is a disjunction of the form  $A \text{ OR } B$ . When applying the All-Edges criterion do you...
1. Have to trigger the edge only once, despite the disjunction?
  2. Have to trigger the edge multiple times, e.g., for several combinations of truth values of A and B?

**Question 2 [5 marks]**

Using an example control flow graph of your choice, illustrate why the Simple-Round-Trip criterion **does not subsume** the All-Nodes criterion. Justify your answer.

**Question 3 [5 marks]**

Consider the graph as specified below:

- There are seven nodes, numbered 1 to 7
  - The edges are the following: (1,2), (2,7), (2,3), (3,4), (3,5), (4,6), (5,6), (6,2)
  - Nodes 1 and 6 define variable x
  - Nodes 4, 5 and 6 use variable x
  - On node 6, the use takes place before the definition
- a) Draw the control flow graph annotated with data flow information. (1 mark)
- b) What are the test requirements for the All-Definitions criterion? (4 marks)

**Question 4 [5 marks]**

White-box integration testing criteria based on coupling and data flow information of arguments and parameters focus on last definitions (before call and before return) and first uses (in callee and after call). Why not considering all the definitions and uses of arguments and parameters in both the caller's and callee's control flow graph? Justify.

**Question 5 [5 marks]**

Consider the following piece of code (the programming language being used is not relevant). Build the corresponding control flow graph: condensed form, number nodes after line numbers.

```
1. i = 0;
2. n=4;
3. while (i<n-1) do
4.   j = i + 1;
5.   while (j<n) do
6.     if A[i]<A[j] then
7.       swap(A[i], A[j]);
8.     else
9.       do-something-else
10.    end do;
11.   i=i+1;
12. end do;
```

**Question 6 [5 marks]**

What are the principles of a bottom-up integration approach?  
Why is bottom-up considered better than big-bang?

**Question 7 [5 marks]**

What is inheritance context coverage?