

# CSCB07 – Software Design

## Exercises related to Software Testing

---

1. Given a boolean expression and a test suite consisting of three test cases as follows:

Expression:  $((a > b) \vee c) \wedge (x < y)$

Test cases:  $(a=2, b=1, c=false, x=4, y=5), (a=1, b=2, c=false, x=4, y=5), (a=1, b=2, c=true, x=5, y=4)$

Indicate whether active clause coverage is satisfied by the test suite. If not, add as many tests as needed to achieve that. Assume we are considering the general case of active clause coverage where the minor clauses don't need to have the same values when the major clause is true as when it is false.

2. Given the same boolean expression as that of question 1, provide a test suite that satisfies clause coverage but not predicate coverage.

3. Given the following code:

```
public int indexOf(int x, int [] A){
    int i=0;
    while(i<A.length){
        if(A[i] == x)
            return i;
        else
            i++;
    }
    return -1;
}
```

- a. Draw the control flow graph.
- b. What is the minimum number of test cases required to satisfy node coverage? Provide an example of such tests.
- c. What is the minimum number of test cases required to satisfy edge coverage? Provide an example of such tests.
- d. Regarding the control flow graph of part a, is there a test suite that satisfies node coverage but not edge coverage? If yes, provide an example. Otherwise, explain why there is no such test suite.

4. The following code is supposed to return the index of the first occurrence of x in A if A contains x and -1 otherwise. However, there is a fault in the code (as shown in the highlighted line).

```
public int indexOf(int x, int [] A){
    int i=0;
    while(i<A.length){
        if(A[i] == x)
            return i;
        else
            i = i + x; //fault, should be i = i + 1;
    }
    return -1;
}
```

Using the RIPR model, indicate for each of the following test cases the conditions that are satisfied.

Test case	Reachability	Infection	Propagation	Revealability
int [] A = {3,8,2}; assertTrue(indexOf(8,A) == 1);	Y	Y	Y	Y
int [] A = {3,8,2}; assertTrue(indexOf(3,A) == 0);	N	N	N	N
int [] A = {3,8,2}; assertTrue(indexOf(1,A) == -1);	Y	N	N	N
int [] A = {3,8,2}; assertTrue(indexOf(2,A) == 2);	Y	Y	N	N
int [] A = {3,2,2}; assertTrue(indexOf(2,A) != -1);	Y	Y	Y	N