

*Family Name*_____

Given Name _____

*Student ID*_____

Curtin University
Department of Computing

Software Engineering Testing
Semester 2, 2017

Mid-Semester Test

Instructions to Students:

- Time allowed: 1 Hour
- This is a closed book Test.
- Total Marks allocated is 60.
- This paper contains 10 pages including the cover page and one rough work page
- Answer all 3 questions.
- Assumptions (if any) should be listed clearly when answering Test questions

Write all answers in the spaces provided.

Question One

(Total 24 marks)

- (I) State whether True OR False (write in the space provided).
(10 X 1 mark each = Total 10 marks for (I))
- (a) Projects that begin test activities after implementation is complete produce reliable software. _____
 - (b) It's possible to satisfy General active clause coverage without satisfying predicate coverage. _____
 - (c) Data flow techniques can be applied to cover activity graphs. _____
 - (d) Unit testing involves only those characteristics that are vital to the performance of the unit under test. _____
 - (e) Any path can be composed by concatenating prime paths. _____
 - (f) Dead code makes it impossible to achieve node coverage. _____
 - (g) Defs and uses for the same variable may appear on the same control flow graph node. _____
 - (h) If test set T1 achieves a higher coverage level than T2 on a set of test requirements (TR), then T1 will detect more defects than T2. _____
 - (i) Verifying the problem statement (or just verification), means checking the correspondence between the problem we have stated and the demands of the real world. _____
 - (j) Predicate coverage helps to evaluate all the clauses. _____

(II) Answer the following questions based the code fragment given below.

(Total 14 marks for (II))

```
class Vehicle implements Cloneable
{
    private int x;
    public Vehicle (int y) { x = y;}
    public Object clone()
    {
        Object result = new Vehicle (this.x);
        // Location "A"
        return result;
    }
    // other methods omitted
}
class Truck extends Vehicle
{
    private int y;
    public Truck (int z) { super (z); y = z;}
    public Object clone()
    {
        Object result = super.clone();
        // Location "B"
        ((Truck) result).y = this.y; // throws ClassCastException
        return result;
    }
    // other methods omitted
}
// Test: Truck suv = new Truck (4); Truck co = suv.clone()
//      Expected: suv.x = co.x; suv.getClass() = co.getClass()
```

(a) Explain what is wrong with the given code. Describe the fault precisely by proposing a modification to the code. (4 marks)

- (b) If possible, give a test case that does not execute the fault. If not, briefly explain why not. **(2 marks)**

- (c) If possible, give a test case that executes the fault, but does not result in an error state. If not, briefly explain why not. **(4 marks)**

- (d) If possible give a test case that results in an error, but not a failure. If not, briefly explain why not. Hint: Don't forget about the program counter. **(1 marks)**

- (e) In the given code, describe the first error state. Be sure to describe the complete state. **(3 marks)**

Question Two

(Total 26 marks)

(I) Consider the given code and answer the following questions. **(12 marks)**

```
public void Test (int x){  
    if (x%2==0)  
        System.out.println("a is even");  
  
    for (int i=1; i<x; i++)  
    {  
        System.out.println(i);  
        if (i>50)  
        {  
            Break;  
        }  
    }  
}
```

(a) Draw the graph and label it using numbers.

(3 Marks)

(b) List minimal test set that satisfies 100% Edge Coverage.

(1 Mark)

(c) List the test requirements for Edge-Pair Coverage.

(2 Marks)

(d) List a minimal test set that satisfies 100% Edge-Pair Coverage. **(1 Mark)**

(e) List the test requirements for Prime Path Coverage.

(3 Marks)

(f) List a minimal test set that satisfies 100% Prime Path Coverage. **(2 Marks)**

(II) Consider the graph:

(Total 14 marks for (II))

```
N = { 1, 2, 3, 4, 5, 6 }
N0 = { 1 }
Nf = { 6 }
E = { (1, 2), (2, 3), (2, 6), (3, 4), (3, 5), (4, 5), (5, 2) }
def(1) = def(4) = {x}
use(3) = use(5) = use(6) = {x}
```

Also consider test paths t_1 and t_2 given below:

```
t1 = [1, 2, 3, 5, 2, 6]
t2 = [1, 2, 3, 4, 5, 2, 6]
```

(a) Draw the appropriate graph.

(2 marks)



(b) Identify the 6 du-paths with respect to x. Please list in sorted order. **(3 marks)**

(c) Fill in the following table with respect to direct du-tours and du-tours with sidetrips. If a test path tours a du-path directly, you do not need to analyze whether it also tours the du-path with a sidetrip. **(3 marks)**

Test Path	Du-Paths Toured Directly	Du-Paths Toured With Sidetrips
t_1		
t_2		

(d) Which du-path is not toured, either directly or indirectly? **(2 marks)**

(e) Using the given test paths, give a minimal test set that satisfy All-Defs (Direct Tours Only). **(2 marks)**

(f) Does the test set $\{t_1, t_2\}$ satisfy All-Uses (Direct Tours Only)? **(2 marks)**

Question Three

(5X2 = Total 10 marks)

Consider the given code and test cases to answer following questions:

```
public void test(int a, int b)
{
    if(a%b==2 || a>b)
        system.out.println("a is valid")
    else
        system.out.println("invalid")
}
```

Test case t1: (a=12, b=10)

Test case t2: (a=2, b=4)

Test case t3: (a=3, b=1)

Test case t4: (a=6, b=6)

Use the above given test cases to answer following questions. Identify the minimal test set for, 100%

(a) Predicate Coverage

Answer:

(b) Clause Coverage

Answer:

(c) Combinatorial Coverage

Answer:

(d) General Active Clause Coverage
Answer:

(e) Restricted Active Clause Coverage
Answer:

END OF TEST PAPER

Please restrict rough work below this line:
