Curtin University

Dept of Computing

End of Semester 2 Examinations
November 20XX

Internal Students

Unit Name:	Software Engineering Testing CMPE4001
Unit Number & Version:	308716

Duration: 2 hrs preceded by a 10 minute reading period. Supervisor will indicate when answering of exam may begin. If you wish to make notes, please use the back of your exam paper, or on the edge columns.

Total Marks: 100 marks.

Aids to be supplied by the University:	None
Aids to be supplied by the Candidate:	None

Calculator: **Not Allowed**

THIS IS A CLOSED BOOK EXAM

Mobile phones or any other devices capable of communicating information are prohibited from use during examinations.
Electronic Organisers/ PDAs or any other devices capable of storing text or other restricted information are prohibited.
Calculators are prohibited.

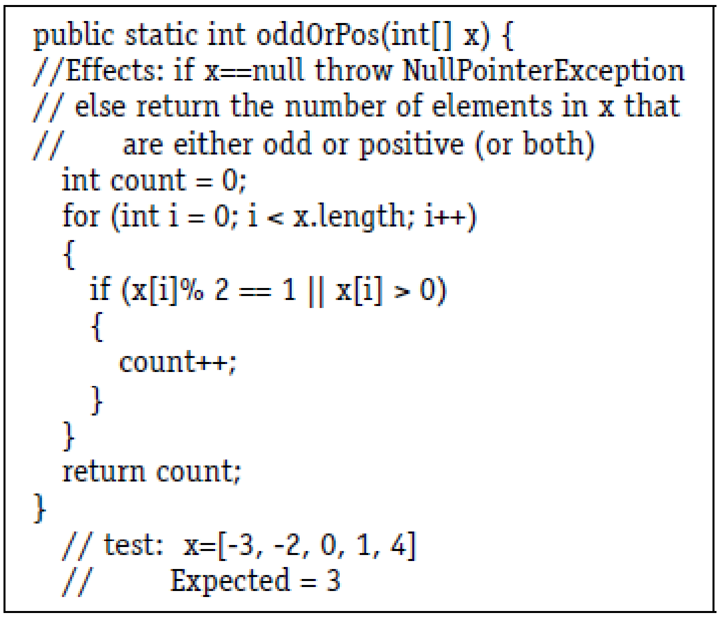
Any breaches of this policy will be considered cheating and appropriate action will be taken as per University policy

GENERAL INSTRUCTIONS:

- **This paper contains FOUR (4) questions:**
    - **Question One is worth 20 marks**
    - **Question Two is worth 30 marks**
    - **Question Three is worth 30 marks**
    - **Question Four is worth 20 marks**

QUESTION ONE **Worth 20 Marks.**

(I) Consider the `oddOrPos ( )` method below. It also includes a test case that results in failure. Answer the following questions based on the given method. **(Total 18 marks)**



```
public static int oddOrPos(int[] x) {
//Effects: if x==null throw NullPointerException
// else return the number of elements in x that
//      are either odd or positive (or both)
  int count = 0;
  for (int i = 0; i < x.length; i++)
  {
    if (x[i]% 2 == 1 || x[i] > 0)
    {
      count++;
    }
  }
  return count;
}
  // test:  x=[-3, -2, 0, 1, 4]
  //        Expected = 3
```

(a) Identify the fault. ***(3 marks)***
(b) Identify a test case that does not execute the fault. ***(3 marks)***
(c) Identify a test case that executes the fault, but does not result in an error state. ***(3 marks)***
(d) Identify a test case that results in an error, but not a failure (Don't forget about the program counter (PC)). ***(3 marks)***
(e) For the given test case, identify the first error state. Be sure to describe the complete state. ***(3 marks)***
(f) Fix the fault and verify that the given test now produces the expected output. ***(3 marks)***

(II) List two factors that would help a development organization move from Beizer's testing level 2 (*testing is to show errors*) to testing level 4 (*a mental discipline that increases quality*)? ***(2 marks)***

QUESTION TWO IS ON THE NEXT PAGE

QUESTION TWO **Worth 30 Marks.**

(I) Consider the given code and answer the following questions. **(Total 11 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 appropriate graph. ***(2 marks)***
(b) List the minimal test set that satisfies 100% Node Coverage. ***(1 mark)***
(c) List the test requirements for Prime Path Coverage. ***(3 marks)***
(d) List a minimal test set that satisfies 100% Prime Path Coverage. ***(3 marks)***
(e) List the test requirements for Complete Path (All unique paths) Coverage. ***(2 marks)***

(II) Edge-Pair Coverage (EPC) ought to impose more test requirements than Edge Coverage (EC). **(Total 9 marks)**
(a) Write a Java method where EPC imposes a test requirement that EC doesn't impose. (You can do this with 3 lines of code.). ***(2 marks)***
(b) Draw the appropriate graph and write out the test requirements for both EPC and EC. ***(3 marks)***
(c) Produce a test set that satisfies EC but not EPC. ***(2 marks)***
(d) Produce a test set that satisfies EPC. ***(2 marks)***

(III) State whether True OR False. **(10 X 1 mark each = Total 10 marks)**
(a) Coverage criterion C_1 imposes an infeasible test requirement t_r on a program P. Coverage criterion C_2 also imposes t_r. There exists a test set that fully meets C_2.
(b) The control flow graph corresponding to a Java method is sometimes not a Single-Entry/Single-Exit graph.
(c) An unreachable program fault can be detected using testing.
(d) To get all-defs coverage, it is sufficient to cover the implicit def at program start for a static field.
(e) Concatenating prime paths also gives a prime path.
(f) Some structural graph coverage criteria subsume some data flow criteria.
(g) A sidetrip is always a simple path.
(h) Fewer paths are semantically possible through a program than are syntactically reachable.
(i) A coupling-du-path contains exactly one use of the coupling variable.
(j) If test path **p** tours subpath **q** with sidetrips, then **p** also tours **q** with detours.

QUESTION THREE IS ON THE NEXT PAGE

QUESTION THREE **Worth 30 Marks.**

(I) Consider the given code and test cases to answer following questions: **(Total 10 marks)**

```
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)
```

Identify the minimal test set for, 100%.

(a) Predicate Coverage. ***(2 marks)***
(b) Clause Coverage. ***(2 marks)***
(c) Combinatorial Coverage. ***(2 marks)***
(d) General Active Clause Coverage. ***(2 marks)***
(e) Restrictive Active Clause Coverage. ***(2 marks)***

(II) Use given predicate to answer the following questions **(Total 10 marks)**

$$p \; = \; a \wedge (\neg b \vee c)$$

(a) Identify the clauses that go with predicate **p**. ***(1 mark)***
(b) Compute (and simplify) the conditions under which each of the clauses determines predicate **p**. ***(3 marks)***
(c) Write the complete truth table for all clauses. Label your rows starting from 1 (based on the definition of Combinatorial Coverage). Include columns for the truth value of the predicate and for the conditions under which each clause determines the predicate. ***(3 marks)***
(d) Identify all pairs of rows from your table that satisfy Correlated Active Clause Coverage (CACC) with respect to each clause. ***(3 marks)***

(III) For the following questions (a) – (c), consider the Finite State Machine (FSM) for a programmable thermostat. Suppose the variables that define the state and the methods that transition between states are: **(Total 10 marks)**

QUESTION THREE IS CONTINUED ON NEXT PAGE

```
partOfDay : {Wake, Sleep}
temp      : {Low, High}

// Initially "Wake" at "Low" temperature

// Effects: Advance to next part of day
public void advance();

// Effects: Make current temp higher, if possible
public void up();

// Effects: Make current temp lower, if possible
public void down();
```

(a) How Many states are there? ***(2 marks)***
(b) Draw and label the states (with variable values) and transitions (with method names). Notice that all of the methods are total. ***(5 marks)***
(c) A test case is simply a sequence of method calls. Provide a test set that satisfies edge coverage on your resultant graph. ***(3 marks)***

QUESTION FOUR IS ON THE NEXT PAGE

QUESTION FOUR **Worth 20 Marks.**

(I) Answer the following questions for the method `intersection( )` below: **(Total 8 marks)**

```
public Set intersection (Set s1, Set s2)
   // Effects:   If s1 or s2 are null throw NullPointerException
   //    else return a (non null) Set equal to the intersection
   //    of Sets s1 and s2

   Characteristic:  Type of s1
      - s1 = null
      - s1 = {}
      - s1 has at least one element

   Characteristic:  Relation between s1 and s2
      - s1 and s2 represent the same set
      - s1 is a subset of s2
      - s2 is a subset of s1
      - s1 and s2 do not have any elements in common
```

a) Does the partition “Type of s1” satisfy the completeness property? If not, give a value for s1 that does not fit in any block. ***(1 marks)***

b) Does the partition “Type of s1” satisfy the disjointness property? If not, give a value for s1 that fits in more than one block. ***(1 marks)***

c) Does the partition “Relation between s1 and s2” satisfy the completeness property? If not, give a pair of values for s1 and s2 that does not fit in any block. ***(2 marks)***

d) Does the partition “Relation between s1 and s2” satisfy the disjointness property? If not, give a pair of values for s1 and s2 that fits in more than one block. ***(2 marks)***

e) If the “base choice” criterion were applied to the two partitions (exactly as written), how many test requirements would result? ***(2 marks)***

(II) Consider the mutant (see line 5′) given in method, `findVal( )` [As shown in Figure below]. **(10 marks)**

```
//Effects: If numbers null throw NullPointerException
//   else return LAST occurrence of val in numbers[]
//   If val not in numbers[] return -1
1. public static int findVal(int numbers[], int val)
2. {
3.    int findVal = -1;
4.
5.    for (int i=0; i<numbers.length; i++)
5′.// for (int i=(0+1); i<numbers.length; i++)
6.       if (numbers [i] == val)
7.          findVal = i;
8.    return (findVal);
9. }
```

QUESTION FOUR IS CONTINUED ON NEXT PAGE

Answer questions (a) – (d) for the mutant.

(a) Find a test input that does **not reach** the mutant. ***(2.5 marks)***
(b) Find a test input that satisfies reachability but **not infection** for the mutant. ***(2.5 marks)***
(c) Find a test input that satisfies infection, but **not propagation** for the mutant. ***(2.5 marks)***
(d) Find a test input that kills mutant `m`. ***(2.5 marks)***

(III) Briefly discuss the notion of Class Integration Test Order (CITO) in Integration Testing. ***(2 marks)***

END OF EXAMINATION PAPER