BITS ID:..……

Birla Institute of Technology & Science, Pilani

AY2017-2018 Semester 1

Software Testing Methods (SS G552)

Comprehensive Examination, Dec 2017 (Closed Book)

Max Marks: 35 **Duration: 180 Minutes**

INSTRUCTIONS:

- **ALL questions are COMPULSORY. The paper has total SEVEN (7) questions on FOUR (4) printed pages.**
- **Comprehensive Examination contributes 35% toward final grade of the course.**
- **ANSWER ALL PARTS OF A QUESTION TOGETHER. FOLLOW THIS INSTRUCTION STRICTLY.**
- **DESPITE the CORRECTNESS of an answer, the QUALITY of the answer is an IMPORTANT EVALUATION criterion. Overwritten answers will not be entertained.**
- **Write your BITS ID on the top corner of the paper. Use of calculator is allowed.**
- **WHILE DESIGNING A TEST SUITE/SET FOR A GIVEN PROBLEM, PROVIDE SUITABLE TEST CASE EXAMPLES WITH APPROPRIATE VALUES.**
- **PROVIDE ALL YOUR ASSUMPTIONS WHILE GIVING JUSTIFICATIONS.**

---

1. **[1M*3=3M]** Differentiate the following terms:
   (a) Static quality attribute and dynamic quality attribute
   (b) Debugging through Backtracking and debugging through program slicing
   (c) Defect and Failure

2. **[1M*2=2M]** Answer the followings questions:
   (a) Using a programming example, shows that unclear requirements can influence the software testing process.
   (b) What is the role of stubs and drivers during unit testing?

3. **[2M*2=4M]** Answer the followings questions:
   (a) Assume that program P' is obtained by modifying an existing program P. Given test set T for P, one of the objective of the regression testing problem is to determine T_r such that successful execution of P' against T_r implies that modified or newly added code in P' has not broken the code carried over from P. Selecting T_r (regression test set) using test minimization is one of the simplest approaches that might lead to a significant reduction in the size of the T_r. Assume that you have chosen the "statement" to be the testable entity of interest for generating regression test set. Show that test minimization based on statement coverage might discard a useful test case that must be included in the regression test set T_r. Use a suitable program example to prove your answer.

(b) Consider a Java class *J* with 16 methods numbered *M1* through *M16*. Let *T'* = {*t1*, *t2*, *t3*, *t4*, *t5*} be the regression test set for class *J'* obtained from test set *T* for class *J*. The methods covered by each test case in *T'* are:

Test case	Methods Covered	Coverage
t1	*M1, M2, M3, M5, M6, M8, M9, M10*	8
t2	*M1, M2, M6, M8, M9, M10, M11, M13, M14*	9
t3	*M1, M2, M3, M5, M6, M7, M8, M9, M10, M14*	10
t4	*M1, M2, M4, M5, M6, M8, M9, M10, M11, M13, M14, M16*	12
t5	*M1, M3, M5, M9, M10, M12, M15*	7

Prioritize the regression test set *T'* using Residual Coverage as the cost criteria. Show all the steps used while prioritizing the regression test set.

4. [3M*2=6M] Answer the followings questions:

(a) Consider two programs *P* and *P'* as shown below where *P'* is obtained by modifying program *P*:

```
Program P                          Program P'
int main() {                       int main() {
1. int a, b, mul, div;             1. int a, b, mul; float div;
2. mul = 0;                        2. mul = 0;
3. div = 0;                        3. div = 0;
4. input (a, b);                   4. input (a, b);
5. if (a < b){                     5. if (a < b)
6.   mul = a*b;                    6.   mul = a*b;
7.   output (mul);}                7.   output (mul);}
   else {                             else {
8.   div = a/b;                    8.   div = a/b;
9.   output (div);}                9.   output (div);}
}                                  }
```

Suppose that following test set *T* is used for testing *P*:

$$T = \{t_1 : < a = 2, b = 3 >, t_2 : < a = 4, b = 1 >, t_3 = < a = 2, b = 2\}$$

Generate required minimal regression test set using execution traces. Mention all the steps used for generating the regression test set. Vague justification or missing/incomplete steps are rewarded with zero marks.

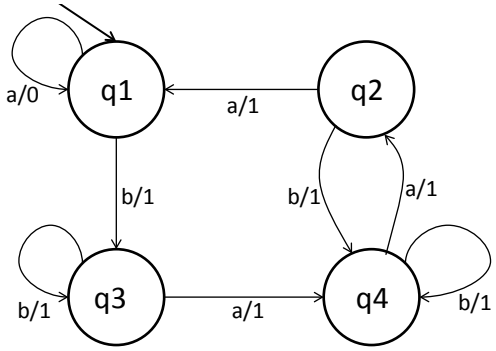
(b) Consider the following program:

```
int foo(int a, int b){
1.  while (a!=b){
2.     if (a>b) then
3.        a=a-b;
4.    else b=b-a;
5.  }
6. return a; }
```

Generate step-by-step a minimal test suite that ensures statement coverage. Does the test suite generated by you ensure branch coverage? Justify your answer. To avoid any ambiguity, justify your statement and branch coverage criteria clearly.

5. **[4M]** Akshay Supermarket planned to offer a discount of 5% for all customers. Since the supermarket allows an option of having membership card for BITS staff and students. Besides a discount of 5% to all customers, such membership card holders can avail either an additional discount of 3% or earn Bonus Points (1 Bonus Point for every INR 25 spent). The card holder has to choose any one of the two additional discounts while making the purchase. Customer without membership card can avail additional offer of 3% discount only if they spend more than or equal to INR 500 per transaction. Based on the above given information, design a minimal test suite using decision table testing. Justify your test suites. Zero marks will be awarded with no/incomplete justification.

6. **[3+3+1=7M]** Let *M=(X, Y, Q, q1, δ, O)* be a minimal and complete FSM where *X = {a,b}, Y = {0,1}* and *Q = {q1,q2,q3,q4}*, Figure below shows the state transition function and output function. Construct *k*-equivalence partitions, distinguishing sequences for each pair of states, and characterization set for the given FSM. Show all the steps used while constructing the *k*-equivalence partitions, distinguishing sequences, and characterization set. Zero marks will be awarded with no/incomplete justification of your answer.



7. **[6+3=9M]** Suppose that a BITS hostel has two types of rooms: Non-AC and AC. The hostel room rent depends on the credit points (CPs) earned by the students in the past year through his/her co-curricular activities. The range for credit points is [0, 50]. You are asked to design a program that takes room-type (in an enumerated data type) and credit points (in an integer variable) as input, and gives the following output:
   (i) If room-type = Non-AC and $35 < CP \leq 50$, rent is INR 1000 pm
   (ii) If room-type = AC and $35 < CP \leq 50$, rent is INR 1500 pm
   (iii) If room-type = Non-AC and $15 < CP \leq 35$, rent is INR 1200 pm
   (iv) If room-type = AC and $15 < CP \leq 35$, rent is INR 1800 pm
   (v) If room-type = Non-AC and $0 \leq CP \leq 15$, rent is INR 1600 pm
   (vi) If room-type = AC and $0 \leq CP \leq 15$, rent is INR 2200 pm

Further suppose that *S* runs on a PC-based platform and prints the output as a printed *Voucher* using a local or a networked printer. The operating system can be Windows7 or Windows8. *S* supports only two browsers: Mozilla and Chrome.

(a) Identify the input domain and construct equivalence classes. Given the identified equivalence classes based on input domain, design a test set using Strong Normal (SN) and Strong Robust (SR) equivalence testing. Show all the steps with proper justifications and mention relevant test suite example. Zero marks will be awarded for no/incomplete justification.

(b) Using combinatorial design, construct minimum set of test configurations based on the above information. Show all the steps used for constructing test configurations.

---------- End of paper ----------