**SE/CS 6367 & SYSM 6310: Software Testing, Validation and Verification**

**Summer 2020**

4:00 – 5:00 pm/online

Tuesday, June 30, 2020

PRINT your name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

UTD ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Write your answer in the space provided following each question. Make it ***concise*** and ***to-the-point***. Providing a simple answer without appropriate explanation or something that is completely irrelevant to the question will not only receive zero point but also result in ***an additional penalty*** from your total score of this exam.

**IMPORTANT:**This is an individual assignment. **All answers should be your own work and in your own words. No answer should be looked up or copied from the internet or any other source (besides the book/notes).** Any indication of cheating or scholastic dishonesty will be passed on to the Dean of Students. If you run into ANY technical difficulties it is imperative that you notify your instructor immediately so that the technical difficulty can be rectified.

1. Many students feel that they have higher confidence in the *correctness* of a program that has been tested with many test cases and conclude “the more a computer program has been tested, the more reliable it is.” (10 points)

Agree or Disagree.

Explain your reasoning.

**A simple YES/NO answer without explanation receives zero point.**

1. Consider the following program that is supposed to check if the input data item is in the range 0 to 100, inclusive

int check(x);

int x;

{

if ((x>=0 )&& (x<=200))

check=true;

else check=false;

}

1. Find a test set T which is adequate with respect to the **block coverage** and **does not** reveal the error (5 points)
2. Find a test set T which is adequate with respect to the **block coverage** and **reveals** the error (5 points)
3. Find a test set T which is adequate with respect to the **simple condition coverage** and **does not** reveal the error (5 points)
4. Find a test set T which is adequate with respect to the **simple condition coverage**   
   and **reveals** the error (5 points)
5. *ExcellentHotel.com* is an online hotel booking website. When a user is ready to check-out, the amount which he/she has to pay is computed as follows:

* Membership discount:
  + Regular users 🡸 no discount
  + Silver membership holders 🡸 20% discount for participated hotels
  + Gold membership holders 🡸 30% discount for participated hotels
  + Platinum membership holders 🡸 40% discount for participated hotels
* Free nights:
  + Gold membership holders with more than 200 reward points 🡸 one free night
  + Platinum membership holders with more than 100 reward points 🡸 one free night
* More for less:
  + books more than three nights 🡸 5% off
  + books more than seven nights 🡸 15% off
* Cancellation policy:
  + cancelled seven days before the checking date 🡸 full refund
  + cancelled within three days before the checking date 🡸 50% refund
  + cancelled within 24 hours before the checking date 🡸 no refund
  + platinum membership holders cancel at any time before the   
    checking date 🡸 full refund
* Special offers:
  + 20% off on participated hotels on Labor Day, Thanksgiving, and Christmas
  + book two nights get one free night on Valentine’s Day, Easter, and Memorial Day
  + 30% off for new customers
  1. Draw a mind map based on the above specification (15 points)
  2. Generate a set of 10 test cases and explain which techniques you have used (5 points)
  3. Explain how to generate additional test cases to improve the test suite in b) and explain the rationale and techniques used for such an enhancement (10 pints)

1. Explain how test cases generated using BVA (Boundary Value Analysis), decision coverage, simple condition coverage, and multiple condition coverage can help programmers test the predicate at s7. Give test cases in terms of *x*, *y*, *a* and *b*. Do not forget to **emphasize the difference** between these techniques.

*s*1 if (*w* = = 2)

*s*2 *x* = 3*a* + *b*

*s*3 *y* = 5*c* − 2*e*

*s*4 else

*s*5 *x* = 3*a* − *b*

*s*6 *y* = 5*c* + 2*e*

*s*7 if ((*x* = = *y*) || (*a* ≤ *b*))

*s*8 print (*x*, *y*)

*s*9 else

*s*10 print (*a*, *b*)

* 1. BVA (Boundary Value Analysis) (5 points)
  2. Decision coverage (5 points)
  3. Simple condition coverage (5 points)
  4. Multiple condition coverage (5 points)

1. **Test case prioritization for regression testing**

The following table shows how each function in a program (*f*1 to *f*6) is executed by a test case (*t*1 to *t*6) such that a “●” indicates that a function is executed by the corresponding test case.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *f*1 | *f*2 | *f*3 | *f*4 | *f*5 | *f*6 |
| *t*1 | ● | ● | ● |  |  |  |
| *t*2 |  | ● |  |  |  |  |
| *t*3 |  |  |  |  | ● |  |
| *t*4 |  |  | ● |  |  |  |
| *t*5 | ● |  |  |  |  | ● |
| *t*6 |  |  |  | ● |  |  |

1. We define a test case to be “*essential*” if and only if a function can only be executed by that test case. Find out all the essential test cases. (8 points)

1. Assume the program under test has *m* functions and *n* test cases.   
   The “*irreplaceability*” of a test case is defined as





Compute the irreplaceability of each test case. (8 points)

|  |  |
| --- | --- |
|  | *irreplaceability* |
| *t*1 |  |
| *t*2 |  |
| *t*3 |  |
| *t*4 |  |
| *t*5 |  |
| *t*6 |  |

1. Based on the information in (a) and (b), explain how to prioritize test cases. (8 points)

1. Redo the prioritization using a greedy approach based on both *irreplaceability* and the *execution cost* of each test case listed below). (8 points)

|  |  |
| --- | --- |
|  | Cost |
| *t*1 | 8 |
| *t*2 | 1 |
| *t*3 | 2 |
| *t*4 | 2 |
| *t*5 | 1 |
| *t*6 | 0.5 |

1. What are other factors that should also be considered for test case prioritization to reduce the cost of regression testing? (8 points)