

## Homework 6, ECE 590 & CS320 Software Reliability

Due Date: 2/21/2024 by 11:59 pm

### **Learn the usage of the recovery block for multi-version sorting – Use of Design diversity:**

1. Come up with a primary variant and an alternate variant using two different sorting algorithms of your choice from bubble sort, selection sort, insertion sort, quicksort, heapsort, etc., and implement them.
2. Design and implement an acceptance test for the sorting algorithm.
3. Treat each sorting routine variant as a block for the purpose of recovery and implement the recovery block scheme.
4. Use a method of logging, such as “printf()”, to track down the calling history of different blocks.
5. Check whether the second sorting routine variant block is ever called. If the second block is not called at all, you can manually insert one or two bugs in your code so as to force the execution of the second variant. A program with manually inserted bugs is known as a mutant. It is not hard to obtain a mutant. The easiest way is to perturb an arithmetic operator, such as replacing “+” by “\*”, and “-” with “/”. You can try several mutants to see the differences.
6. You may use C/C++, Java, or Python to finish this assignment.

**Please hand in the codes you implement along with a writeup in PDF to demonstrate how to run your code, and how you achieve the recovery block.**

Hint: The recovery block scheme is:

Ensure	<i>acceptance test</i>
By	<i>primary variant</i>
Else by	<i>first alternate variant</i>
Else by	<i>second alternate variant</i>
.....	
Else by	<i>last alternate variant</i>
Else fail	