

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

1. Please implement two variants of sorting algorithm, such as bubble sort, selection sort, insertion sort, quicksort, heapsort, etc.
2. You will need to implement an acceptance test for sorting.
3. Treat each sorting algorithm 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 variant block will be called. If the second block is not called at all, you can manually insert one or two bugs in your code to check again the calling history. 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 readme to demonstrate how your codes work.

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	