Lecture 1 - In Class Exercise

Goal: Think about software testing in a more rigorous way than you may be used to. The exercise also hints at the strong relationship between specification clarity, faults, and test cases.

1 Explore “Testing”

Instructions: Work with your neighbors in groups of 2

Consider the following Java method:

```java
public static Vector union (Vector a, Vector b) {
   Vector result = new Vector (a); // get all of a’s elements
   Iterator itr = b.iterator();
   while (itr.hasNext()) {
      Object obj = itr.next();
      if (!a.contains (obj)){
         result.add (obj);
      }
   }
   return result;
}
```

1. The method should return a Vector of objects that are in either of the two argument Vectors. Upon reflection, you may discover a variety of defects and ambiguities. In other words, ample opportunities for faults exist. Identify as many possible faults as you can.
2. Create a set of test cases that you think would have a reasonable chance of revealing the faults you identified above. Document a rationale for each test in your test set. If possible, characterize all of your rationales in some concise summary. Run your tests against your implementation.
3. Rewrite the method signature to be precise enough to clarify the defects and ambiguities identified earlier. You might wish to illustrate your specification with examples drawn from your test cases.