SE333-433: Class Activity #2

The intention of these class activities are to encourage you to think of testing in a more rigorous way than you may be used to. These exercises also hint at the strong relationship between specification clarity, faults, and test cases. Please submit an image of your notes to d2l under class activity 2.

1. Below is a faulty program. Answer the following questions about the program.

```
public int findLast (int[] x, int y){
     for (int i=x.length-1; i > 0; i--){
          if (x[i] == y) {
                return i;
          }
     }
          return -1;
}
```

(a) Explain what this code is supposed to do? Explain what is wrong with the given code?

(b) Give a test case that does not execute the fault.

(c) Give a test case that executes the fault, but does not result in an error state.

(d) Give a test case that results in an error state, but not a failure.

(e) Give a test case that result in failure.

2. Below is a faulty program. Answer the following questions about the program.

```
public static int oddOrPos(int[] x){
     int count = 0;
     for (int i = 0; i < x.length; i++){
          if (x[i]%2 == 1 || x[i] > 0) {

               count++;

          }

      }

     return count;

}
```

(a) Explain what this code is supposed to do? Explain what is wrong with the given code?

(b) Give a test case that does not execute the fault.

(c) Give a test case that executes the fault, but does not result in an error state.

(d) Give a test case that results in an error state, but not a failure.

(e) Give a test case that result in failure.

3. Consider the following class. There is OO faults taken from Joshua Bloch's Effective Java, Second Edition. Answer the following questions about each.

```
class Vehicle implements Cloneable {

      private int x;
      public Vehicle (int y) { x = y;}
      public Object clone() {

            Object result = new Vehicle (this.x);
            // Location "A"
            return result;
      }
      // other methods omitted
}

class Truck extends Vehicle {
      private int y;
      public Truck (int z) { super (z); y = z;}
      public Object clone() {
            Object result = super.clone();
            // Location "B"
            ((Truck) result).y = this.y; // throws ClassCastException
            return result;
      }
      // other methods omitted
}
```

(a) Explain what this code is supposed to do? Explain what is wrong with the given code?

(b) Give a test case that does not execute the fault.

(c) Give a test case that executes the fault, but does not result in an error state.

(d) Give a test case that results in an error state, but not a failure.

(e) Give a test case that result in failure.