Homework1- (individual) **SE 4367: Software Testing, Verification, Validation and QA**

Student Name: Alex Lundin
Class and Section SE 4367.0U1
Total Points (Out of 100 points) _____

Instruction:

1. Answer to the problem on a **PDF** file **(PDF file only)**
2. Submit the **PDF** file to eLearning before the due date

1. (5 points) Please **subscribe** to the Class Discussion Board for important announcements, questions, and updates.

2. (75 points) Below are four faulty programs. Each includes test inputs that result in failure. Answer the following questions about each program.

```
/**
 * Find last index of element
 *
 * @param x array to search
 * @param y value to look for
 * @return last index of y in x; -1 if absent
 * @throws NullPointerException if x is null
 */
public int findLast (int[] x, int y)
{
   for (int i=x.length-1; i > 0; i--)
   {
      if (x[i] == y)
      {
         return i;
      }
   }
   return -1;
}
// test:  x = [2, 3, 5]; y = 2; Expected = 0
// Book website: FindLast.java
// Book website: FindLastTest.java
```

```
/**
 * Find last index of zero
 *
 * @param x array to search
 *
 * @return last index of 0 in x; -1 if absent
 * @throws NullPointerException if x is null
 */
public static int lastZero (int[] x)
{
   for (int i = 0; i < x.length; i++)
   {
      if (x[i] == 0)
      {
         return i;
      }
   }
   return -1;
}
// test:  x = [0, 1, 0]; Expected = 2
// Book website: LastZero.java
// Book website: LastZeroTest.java
```

```
/**
 * Count positive elements
 *
 * @param x array to search
 * @return count of positive elements in x
 * @throws NullPointerException if x is null
 */
public int countPositive (int[] x)
{
   int count = 0;
   for (int i=0; i < x.length; i++)
   {
      if (x[i] >= 0)
      {
         count++;
      }
   }
   return count;
}
// test:  x = [-4, 2, 0, 2]; Expcted = 2
// Book website: CountPositive.java
// Book website: CountPositiveTest.java
```

```
/**
 * Count odd or postive elements
 *
 * @param x array to search
 * @return count of odd/positive values in x
 * @throws NullPointerException if x is null
 */
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;
}
// test:  x = [-3, -2, 0, 1, 4]; Expected = 3
// Book website: OddOrPos.java
// Book website: OddOrPosTest.java
```

(a) Explain what is wrong with the given code. Describe the fault precisely by proposing a modification to the code.
(b) If possible, give a test case that does not execute the fault. If not, briefly explain why not.

(c) If possible, give a test case that executes the fault, but does not result in an error state. If not, briefly explain why not.
(d) If possible give a test case that results in an error state, but not a failure. Hint: Don't forget about the program counter. If not, briefly explain why not.
(e) For the given test case, describe the first error state. Be sure to describe the complete state.

3. (20 points) Download the facts.zip file which contain the source files. It is a web application written in Java with a combination of Java servlets, Java server pages, JavaScript, and XML. We will be using it for future evolution and testing assignments. You will not need to be an expert in web apps to do these assignments, but you will need to understand how this small application works. For this assignment, all you need to do is download the files and examine them to see how well you can understand the code. You can download the files anywhere, but I recommend putting them in a repository like GitHub. **Submit a screen shot that shows where you put the facts files.**