CSCI 4422 | CSCI 5599
HW 01 - Preliminary Concepts

Solution Key

Assigned: January 13, 2020
Due: January 19, 2020 @ 23:00h

Questions (20 pts)

Answer the following questions. All assignments are due **before** class on the due date.

1. (5 pts). Introduce yourself in the forum on Moodle. Embed a picture inline. Then tell us your name, major, where you are from, your career goals, why you are taking this class, and what you expect to learn. Post a followup discussion item to my introduction note. **This will be graded by checking moodle.**

To grade this simply check if: * (1 pt) They have created a followup post to my posting. * (1 pt) They have posted a picture inline, along with their name, major, and where they are from. * (1 pt) They have added their career goals * (1 pt) They posted why they are taking the course. * (1 pt) They posted what they expect to learn.

2. (15 pts) Ammann & Offutt, edition 2, Exercises Chapter 1, Number 5. Answer questions (a) through (f) for findLast() only. **Submit your answers as a PDF to moodle**.

```java
/**
 * 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
```

a) Explain what is wrong with the given code. Describe the fault precisely by proposing a modification to the code.

   The condition in the for loop, `i > 0`, precludes the ability to see values at index 0. Thus the change should be to make the condition `i >= 0`

b) If possible, give a test case that does **not** execute the fault. If not, briefly explain why not.

   test: x = **null**, y = 0; Expected = NullPointerException

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.

   test: x = [1, 2, 3], y = 2; Expected = 1

d) If possible give a test case that results in an error, but **not** a failure. If not briefly explain why not. Hint: Don't forget about the program counter.

   test: x = [1, 2, 3], y = 4; Expected = −1

e) For the given test case, describe the first error state. Be sure to describe the complete state.

   Error State: x = [2, 3, 5], y = 2, i = 0, PC = "`i > 0`"

f) Implement your repair and verify that the given test now produces the expected output. Submit a screen printout or other evidence that the new program works.

```
public class Test {

    public static void main(String[] args) {
        Test t = new Test()
        int x[] = {2, 3, 5}
        int ndx = t.findLast(x, 2)
        assert(ndx == 0)
    }

    /**
     * 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
}
```

Output: