PGCert IT: Programming for Industry

Lab 10: Recursion

Exercise One: Simple Recursion Exercises

For this exercise, answer the following problems on paper. Try to figure out the answer by hand, rather than just typing the code into your IDE and writing down the output. This is important practice for possible upcoming test questions!

1. Consider the code below. What is the return value when `foo(4)` is executed?

```
private int foo(int x) {
    if (x <= 1) {
        return 1;
    }

    return x * foo(x - 1);
}
```

2. Consider the code below.

```
private double bar(double x, int n) {
    if (n > 1)
        return x * bar(x, n - 1);
    else if (n < 0)
        return 1.0 / bar(x, -n);
    else
        return x;
}
```

   a. What is the return value when `bar(2, 3)` is executed?

   b. What is the return value when `bar(3, -2)` is executed?

3. Consider the code below. What is the problem with it?

```
private void bad1() {
    System.out.println("This is very good code.");
    bad1();
}
```

4. Consider the code below. What is the problem with it?

```
private int bad2(int n) {
    if (n == 0) {
        return 0;
    }

    return n + bad2(n - 2);
}
```

5. Consider the code below:

```
private int bad3(int n) {
    if (n == 0) {
        return 0;
    }

    return n + bad3(n + 1);
}
```

   a. Write a method call to bad3() that will cause problems.

   b. Write a method call to bad3() that will not cause problems.

Exercise Two: Writing Recursive Methods

For this exercise, complete the recursive methods in the `ExerciseFour` class in the `ex04` package. You may use the provided unit tests to check whether your methods are functioning correctly. Use recursion to complete each of these methods (even if you can think of a non-recursive way to solve them)! The methods you should complete are as follows:

1. Complete the `getSum` method, which should return the sum of all positive integers between 1 and `num` (inclusive).

2. Complete the `getSmallest` method, which should return the smallest element in an array of integers between the given first (inclusive) and second (non-inclusive) indices.

3. Complete the `printNums1` method, which should print out all the integers starting from the provided positive integer `n`, down to 1.

4. Complete the `printNums2` method, which should print out all the integers starting from 1, up to the provided positive integer `n`.

5. Complete the `countEs` method, which returns the number of ‘e’ and ‘E’ characters in the provided `String`.

6. Complete the `fibonacci` method, which should return the n^{th} number in the Fibonacci sequence.

7. Complete the `isPalindrome` method, which should return a `boolean` value indicating whether the provided `String` is a palindrome.