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);
}
```

4! = 24

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?
   2^3 = 8

   b. What is the return value when `bar(3, -2)` is executed?
   3^-2 = 1/9 (0.11111...)

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

```
private void bad1() {
```

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

No base case - will get StackOverflowError

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);
    }
```

Base case only reached when n is positive and even. Otherwise, will get StackOverflowError.

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.

      bad3(1) - if n is positive, base case will never be reached.

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

      bad3(-1) - if n is negative or 0, base case will be reached.