Practice 4.3

Solution:

1)

Soln:

```
public class Linear {

    // Method to calculate factorial using recursion
    public static double factorial(double n) {
        // Base case: if n <= 1, return 1
        if (n <= 1) {
            return 1;
        }
        // Recursive case: n * factorial(n - 1)
        else {
            return n * factorial(n - 1);
        }
    }

    // Main method to test the factorial method
    public static void main(String[] args) {
        // Declare a double variable and assign the value 5.0
        double d = 5.0;

        // Calculate factorial of d
        double fact = factorial(d);

        // Print the result in the specified format
        System.out.printf("Factorial [%.1f] of [%.1f]%n", fact, d);
    }
}
```

2)

Soln:

```
public class NonLinear {

    // Method to calculate Fibonacci number using recursion
    public static double fibonacci(double n) {
        // Base case: if n is less than 2, return n
        if (n < 2) {
            return n;
        }
        // Recursive case: fibonacci(n - 1) + fibonacci(n - 2)
        else {
            return fibonacci(n - 1) + fibonacci(n - 2);
        }
    }

    // Main method to test the fibonacci method
    public static void main(String[] args) {
        // Declare a double variable and assign the default value 5
        double d = 5.0;

        // Check if command-line arguments are provided
        if (args.length > 0) {
            try {
                // Parse the first argument to a double
                d = Double.parseDouble(args[0]);
            } catch (NumberFormatException e) {
                // Handle the case where the argument is not a valid double
                System.out.println("Invalid input, using default value 5.0");
            }
        }
        // Print Fibonacci values for indices from 0 to d (inclusive)
        for (int i = 0; i <= d; i++) {
            double fibValue = fibonacci(i);
            System.out.printf("Fibonacci index [%.1f] value [%.1f]%n", i, fibValue);
        }
    }
}
```

3)

Soln:

```
public static double factorial(double d) {
    if (d <= 1) {
        return 1;
    } else {
        return d * factorial(d - 1);
    }
}
```