

Standard Recursion, Tail Recursion, Loops

Important Dates:

- Assigned: September 11, 2024
- Deadline: September 25, 2024 at 11:59 PM EST

Objectives:

- Students learn to design more complex methods.
- Students understand and describe the differences between standard recursion, tail recursion, and iteration.
- Students understand the direct correspondence between iteration and tail recursive methods.
- Students design methods that call `private` helper methods to solve a problem.

What To Do:

For each of the following problems, create a class named `ProblemX`, where `X` is the problem number. E.g., the class for problem 1 should be `Problem1.java`. Write (JUnit) tests for each method that you design in corresponding test files named `ProblemXTest`, where `X` is the problem number. Additionally, write Javadoc comments explaining the purpose of the method, its parameters, and return value. **Do not round your solutions!**

You must write sufficient tests and adequate documentation.

All problems are listed in *Learning Java - A Test-Driven Approach*. This problem set contains ten required problems, meaning the maximum possible score on this problem set is 100%/100%.

1. Exercise 2.31 [Palindrome determiner]
2. Exercise 2.35 [Hyperfactorials]
3. Exercise 2.36 [Subfactorials]
4. Exercise 2.37 [Collatz conjecture]
5. Exercise 2.44 [ASCII to integer]
Note: you cannot use methods that solve the problem for you, e.g., `Integer.parseInt`, because that defeats the entire point of the problem.
6. Exercise 2.46 [Wordle recreation]
7. Exercise 2.48 [Recreating the substring method]
8. Exercise 2.51 [File name comparison]
9. Exercise 2.58 [Summing numbers in a string]
10. Exercise 2.68 [Rudimentary calculus]