CSDS 293
Software Craftsmanship
2025 Spring Semester

Programming Assignment 1

Due at the beginning of your discussion session on

January 21-24, 2025

Quiz

In addition to the following topics, the quiz syllabus includes any material covered in the lectures:

- Chapter 2: "Software Construction: Building Software" in Code Complete
- Table 3.1, "Choosing Between Iterative and Sequential Approaches", Sections 3.3 and 3.5 in Code Complete
- Section 4.3 in Code Complete

Programming

In this assignment, you will write a method that finds the longest higher suffix in two lists:

```
static <T extends Comparable<? super T>>
List<T> longestHigherSuffix(
      List<T> a, List<T> b, Comparator<T> cmp)
```

A *suffix* of a list is a list containing the last entry (or entries) of the list. A higher suffix is a suffix of list `a` whose elements are greater than or equal to the corresponding element in list `b`. The method is supposed to return the higher suffix of maximum length. Here are some examples:

a	b	Longest higher suffix
0, 2, 4	1, 2, 3	2, 4
1, 2	2, 1	2
2, 4	1, 3, 2, 4	2, 4
1, 2, 3, 4	1, 2, 4	2, 3, 4
2, 1	1, 2, 3	Empty list
1, 3, 2, 4	1, 2, 3, 4	4

To make the assignment more exciting:

- If your CWRU network id ends with a 0, 3, 6, or 9, then your code should use `Iterator`s or `ListIterator`s.
- If your CWRU network id ends with a 1, 4, or 7 then your code can use recursion but cannot use any type of loop (`while`, `do`, `for`, for-each, `Iterator`, `ListIterator`, `Streams`, etc.)
- In all other cases, your code should use `Streams`, but cannot use any other type of loop (`while`, `do`, `for`, for-each, `Iterator`, `ListIterator`, etc.) or recursion, and should avoid programming loops "into" `Stream`.

Optional Work

The following two work items are given as an option and are not required as a part of the assignment.

Your implementation can optionally include a `main`. If you opt to include a `main`, read two strings from standard input and print on standard output their longest higher suffix. The program will convert the string into lists and use the appropriate version of the generic `longestHigherSuffix`. Create your own input data and run your program on it.

Another option is to solve the assignment using generative artificial intelligence, such as ChatGPT of GitHub co-pilot, and compare your solution with the automatically generated one. You should try to answer the following questions about the automatically generated solution:

- Is it correct for all inputs? How can you check its correctness?
- How would you test it?
- Is it readable?
- Is it efficient?
- Can you improve it?

For a more nuanced comparison, it is recommended that artificial intelligence be used only *after* you have solved the assignment.

Canvas Resource

The Course Document page contains links to some of the Java features that are helpful for this assignment, such as collections and streams.

Submission

Submit an electronic copy of your program to canvas.

Grading Guidelines

The first assignment is required but not graded.