EECS 293
Software Craftsmanship
2019 Fall Semester

Programming Assignment 1

Due at the beginning of your discussion session on

September 4-6, 2019

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 smaller prefix in two lists:

```
static <T>
List<T> longestSmallerPrefix(List<T> a,
                             List<T> b,
                             Comparator<? super T> cmp)
```

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

a	**b**	**Longest smaller prefix**
1, 2, 4	1, 2, 3	1, 2
1, 2	2, 1	1

1, 2	1, 3, 2, 4	1, 2
1, 2, 3, 4	1, 2, 4	1, 2, 3
2, 1	1, 2, 3	Empty list
1, 3, 2, 4	1, 2, 3, 4	1

To make the assignment more exciting:

- If your CWRU network id ends with a 0, 3, 6, or 9, then your code should use `Iterators`.
- 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. (In Python, use streams or higher order functions.)

Your implementation can optionally include a `main` but it does not have to. If you opt to include a `main`, read two strings from standard input and print on standard output their longest smaller prefix. The program will convert the string into lists and use the appropriate version of the generic `longestSmallerPrefix`. Create your own input data and run your program on it.

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

Additionally, submit an electronic copy of your program to canvas.

Notes

If you are in a non-Java section, it is your responsibility to adapt these and future specifications to your programming language.

Grading Guidelines

The first assignment is required but not graded.