
ASSIGNMENT 3

CS 474: Object-Oriented Languages and Environments / Spring 2023

Description

In this assignment, you will practice using Java streams and functional programming. There are 10 test cases, worth 10 points each: undergraduate students need to get 70 points total for full credit (test cases 1-7), while graduate students should get all 100 points.

For this assignment, you need to write a class that implements the interface **StreamProcessor**. It includes six methods, which should be implemented as follows:

Problem 1: Given a list of integers, return only the positive numbers from the list.

Problem 2: Given a list of items implementing the **HasId** interface, create a map from their ids to the corresponding items.

Problem 3: Given two arrays of strings, return a set containing the strings that appear in both arrays.

Problem 4: Given a list of lists of strings, return a map from each string to the number of times it appears across all the lists.

Problem 5: Given a list of **Semester** objects, return a set of all the **Course**s within those semesters that had more than 20 students.

Problem 6 (grad students only): given a list of **ClassOrInterfaceDeclarations** (from the JavaParser library from Assignment 1), return a map from each class name C to the set of other classes that have at least one method name in common with C.

Java Streams

Here is a list of some stream operations that may be useful. For more details, see the lecture slides and the [Java Stream API](#).

- Non-terminal operations

- filter
- map
- flatMap

- Stream manipulation

- Stream.of
- Stream.concat

- Terminal operations

- forEach
- toArray
- findFirst / findAny
- collect
 - Collectors.toList
 - Collectors.toSet

○ Arrays.stream

▪ Collectors.toMap

You should use these operations instead of any other control flow (loops, recursive calls, etc.). We will manually check your code to make sure that you haven't used loops or recursion instead of stream operations.

Due Date and Resubmission Policy

This assignment is due on **March 31 2023** (Friday) at **11:59pm CST**.

The code and date used for your submission is defined by the last commit to your Git repository.

Submission and Grading

This assignment is submitted through GitHub. You can check your current grade at any point by submitting your code and checking the autograder. The automatic grade is determined by 10 tests that will check if your project outputs the expected result. Each test is worth 10%. The autograder will **not** check that you have actually used streams and functional programming – we will do that manually after the deadline – so please make sure you're not using loops or recursion to solve the problems!

Errors and Omissions

If you find an error or an omission, please post it on Piazza as soon as you find it.

Academic Integrity

The academic integrity policy described in the syllabus applies to this assignment. You are responsible for writing all the code that you submit. We will use an automatic tool that detects plagiarism on all submitted code, and we will investigate all instances where plagiarism is more than likely.

Please refer to the syllabus for the full academic integrity policy.