

---

# ASSIGNMENT 3

---

CS 474: Object-Oriented Languages and Environments / Fall 2021

## Description

In this assignment, you will have to reimplement a portion of Assignment 1 and Assignment 2 using Java streams and functional programming.

## Correctness – 40%

Your submission of Assignment 3 **should pass the same tests** as your previous submissions of Assignment 1 and Assignment 2 did. This means that the code you wrote is correct, and you did not introduce any error when transforming the code from imperative to functional.

Each test in your submission of both Assignment 1 and Assignment 2 is worth 10%, for a maximum of 40%. As long as you scored 40% between Assignment 1 and Assignment 2, you can get the full correctness grade (e.g., 10% on Assignment 1 and 30% on Assignment 2, you get 40% on Assignment 3 as long as you don't fail tests you passed previously).

If you resubmitted any assignment, you must use your resubmission code.

## Java streams – 60%

You have to rewrite a portion of your submissions, using the Java stream operations described below. Each operation **highlighted** is worth 10%. You will claim each 10% by explaining how you changed your code, see the next section on the screen-cast video submission.

Creating a **tail-recursive method** also counts towards the 60% listed here. This means that you can create one tail-recursive method, plus 5 stream operations below, and get 60%.

- Non-terminal operations
  - filter
  - map
  - flatMap
- Stream manipulation
  - Stream.of
  - Stream.concat
  - Array.stream
- Terminal operations
  - forEach
  - toArray
  - findFirst / findAny
  - collect
    - Collectors.toList
    - Collectors.toSet
    - Collectors.toMap

## Screen-cast

To claim the 60% listed in the previous section, you must submit a screen-cast (**through Gradescope**) where you explain how each stream operation works in the scope of the assignment. **The maximum length for the video is 5 minutes, videos over 5 minutes have a 10% penalty and instructors will stop watching at the 5 minute mark (nothing past that point in the video will be graded).** This video should be a screencast of your IDE open on the code submitted, and you should highlight the code and narrate the purpose of the highlighted code.

Examples of valid reasons:

- Here we use the map operation to turn the stream of Object into a stream of Class. We use a lambda function that <explains lambda function inside map>.
- Here we filter all the elements in the stream with a lambda function that returns true when <explain when>.
- Here we create a stream of <stream type> from an array of <array type>, and then we <use points above to describe operations made on the stream>.

You can record such a video without installing any software by using the following website:

<https://screenapp.io/#/>

## Trivial Operations

It is possible to write stream operations that do nothing, in the sense that they do not change the underlying data in any way and can be discarded. For instance, imagine a submission with the line:

```
Method[] = class.getDeclaredMethods();
```

This line could be replaced with:

```
Method[] = Arrays.stream(class.getDeclaredMethods())
    .map(m -> m) // Maps each element to itself
    .filter(m -> true) // Filters no elements
    .toArray(); // Returns the original array
```

All the stream operations can be considered trivial as they do not change the stream in any way.

Trivial stream operations do not count, and the instructors will not consider them when grading your submission.

Non-trivial stream operation change the stream in either the number of elements or the type of the elements between the first call to stream and the terminal operation on that stream.

## FAQ

***I used Java streams in Assignment 1 and/or Assignment 2. Do I have to rewrite them for Assignment 3?***

No. If you used Java streams in Assignment 1 or Assignment 2, you don't need to rewrite them to claim the points of the respective operations on the screen-cast. You just need to explain them.

***My grades for Assignment 1 and Assignment 2 add to less than 40%. Can I submit Assignment 3?***

The correctness criterion is waived for you. You will get the 40% if your submission for Assignment 3 scores 40% or more, even if you fail tests that your earlier submissions passed. If you reuse your code with less than 40% (e.g., 10% for Assignment 1 and 20% for Assignment 2), then you will have that grade for the correctness part (e.g., 30%).

I strongly suggest that you go over the solutions released and use that as your starting point.

***Can I pass more tests for Assignment 3 than I did for Assignment 1 and 2?***

Yes, as long as you don't fail any test you passed before. No substitutions, if you failed Test 2 but passed Test 9 you cannot now fail Test 9 and pass Test 2.

***My submission for Assignment 1 and 2 is hard to reuse. Can I use the solution released instead?***

Yes.

## Due Date and Resubmission Policy

This assignment is due on **October 23 2021** (Saturday) at **5pm CST**. There is no late policy.

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

To resubmit this assignment, your **original grade** (as defined by the autograder) should be **equal to or higher than 30%**. You can resubmit your assignment until **October 30 2021** (following Saturday) at **5pm CST**.

Together with your resubmission, you will have to submit a written description of what you changed from the original submission (on Gradescope).

## Bonus Points

This assignment has a total of **10% bonus points**, which you can earn by using Piazza as described in the syllabus. Your posts should be public, tagged with the `assignment3` label, and non-anonymous to the instructors to count towards the bonus.

## Submission and Grading

This assignment is submitted through Github, and has an automatic grade component of 40%. **The instructors will check if you fail any test that you passed previously after the submission.** You can check your current grade at any point by submitting your code and checking Travis.

The final grade for the assignment will be the grade of the original submission, for assignments without a resubmission; or the average between the original grade and the resubmission grade, for assignments with a resubmission. The grade of the original submission includes any bonus points.

## Errors and Omissions

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

## Hardcoding and Academic Integrity

Any hardcoding will result in a 0% grade. Hardcoding is when you submit code that detects which test is being run, and simply outputs the expected result. For instance, detecting that test 22 is running, and replacing the usual execution of your submission with `System.out.println("expected result")`.

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.