



Overview

At Commerce Architects there are three activities that software developers will spend the majority of their time on. These are debugging code, refactoring code, and writing new code. To represent these aspects of the job you have been provided with 3 possible homework assignments, each corresponding with one of these activities. Of these 3 options, you are expected to select 2 for completion.

Homework Options

New Code: Grading Application

The Java streams API (`java.util.stream.Stream`) is an example of functional programming in Java. Using this API, create a stream that generates random final exam grades between 0 and 100 using the below calculation to obtain a lognormal distribution of grades. Be sure to clip any grades under 0 to 0 and over 100 to 100.

```
int grade = (int) (100.0d - Math.exp(random.nextGaussian() * 0.25 + 3.5) + 20);
```

Now, for the first 1,000 random grades from the stream above, calculate and display the following, again using Java streams functionality and functional programming as much as possible:

1. Basic statistics like the min, max, and average of the 1000 grades
2. The number of students that received each letter grade (A, B, C, D, F) like this:
Letter grade A: 354 students
(how concise can you make the code that does this?)
3. A count of how many students got a perfect score of 100.

Hints: read up on lambdas and functional interfaces.

Refactoring: Analytics Application

This homework assignment comes in two flavors. The zip file contains a Java directory and a Typescript directory. You only need to complete the homework in one of the two directories, choose whichever language you prefer. The setup instructions and detailed explanation of the requirements are included in README files specific to each language.

You are provided with a working application. This application consumes JSON documents containing employee analytics data, and given an employee name determines their greatest period of productivity. There is a Parser file, a Processor file, and a variety of model files. However, the `Processor::getBestProductivityForEmployee` function is poorly written. This method must be refactored without changing the method signature or the functionality of the code within.

Again, setup, additional instructions, and hints are provided in the README file specific to the language you chose.

Debugging: Calculator Application

You are provided with a Java application containing a single bug. This problem manifests in a couple of different ways but has been reproduced for you in the form of a unit test. Your job will be to identify the cause of the bug and provide a short paragraph that does the following:

- Describe specifically what is causing the problem
- Describe the process you used to identify the problem
- Propose a solution

Setup, additional instructions and hints are provided in the application's README file.