

# Project 2: Building a Global Index

---

DS 730

## Overview

In the second project, you will be working with input, output, classes in the Java Collections Framework and threads. You are expected to investigate the Java library and use the classes and methods in the Collections library as much as possible. This project is an extension of the first project and allows you to work with more advanced multithreading topics.

In short, you are creating a global word index. This differs from the first project in that you need to *send* your results back to the master thread so that the master thread can produce the output. Only the master thread is allowed to produce any output.

## Project Tasks

You must implement the same general concepts as the first project with a few small changes. You are only solving this problem one way using threads. The following are the changes that must be made:

1. The character requirement and page requirement are identical to the first project.
2. Create a word index for each file and store that each index into a single output file. The output should be comma separated with the first line being the heading. The heading must be:

**Word, FileOne, FileTwo, FileThree, etc.**

### Example:

If you have input files called **a.txt**, **b.txt**, **c.txt**, then your heading must be:

**Word, a.txt, b.txt, c.txt**

3. Your outputs will be combined and the output file will look like the following.

Assume the word **cat** is in:

- a.txt on pages 2, 4, 6.
- b.txt on 3,5,7.
- c.txt on 2, 5, 7.

Your output line for cat would be:

cat, 2:4:6, 3:5:7, 2:5:7

Since commas are used to delimit the files, colons will be used to delimit page numbers.

**Example:**

Assume the word **dog** is:

- *in* a.txt on pages 2 and 5.
- *not in* b.txt .
- *in* c.txt on page 8.

Your output line for dog would be:

dog, 2:5, , 8

4. You are welcome to use whatever callback or synchronization solution you want, but you have to make sure that only the master thread does the output. No worker thread is allowed to print anything.

**Note:** I have included a .zip file containing several sample input and output files. You can find this file in the online course. If your program creates a word index that is identical to mine, you likely wrote your program correctly.

## Submitting Your Work

When you are finished, submit a .zip file to the **Project 2 dropbox** that includes:

- Your code.