

# Return of KWIC – Client + Server

This is as close to a “research project” as it gets in this course. There is no provided code or thorough guidelines or even tests. This assignment is 100% figuring it out **on your own**. That is why it’s a 2 week assignment – all intended.

The Client:

Will send lines to the server, which will be circularly shifted and alphabetized by the server.

Wait/listen...

Note that there should not be any file input. You can hard-code any example set of lines to send to the server in the client.

The Server:

Will receive all lines, circularly shift them, alphabetize them, and send them back to the client.

The Client:

Receive all the lines that have been shifted and alphabetized, in order, then sysout them all in order.

Note that there should not be any file output.

**Academic Honesty: I will not hesitate to give 0s to any students that work together. This is an individual assignment! If you find code on the internet (or AI) and copy-paste it, and other students do the same, you will all get caught in the plagiarism report! Use the internet to learn, not copy paste!**

Please see thorough submission instructions on next page!

For extra credit:

Make the Server multithreaded. You must demonstrate multiple clients using the server at the same time with different KWIC inputs! You must demonstrate visual proof (not just code) that the clients are being served at the same time (how you do this is up to you)! Different clients should use different hard-coded inputs to make it clear that they are different.

Rough grading rubric:

- Remember: functional code is the bare minimum for a passing grade, not a good grade. This is your opportunity to demonstrate what you have learned in this course when it comes to software design.
- Video quality, especially audio (please don’t destroy our ear drums)
- Clean Code
- Design principles (encapsulation, coupling, cohesion, SRP, etc)
- Formatting

# Special Submission Instructions

You will be required to create a video (with your voice, but not your face) of your finished product. The video should include:

- A walkthrough of your finished product, demonstrating its functionality.
- Clearly demonstrate the client-server design working correctly, as described above.
- A thorough walkthrough of your code and how it works.

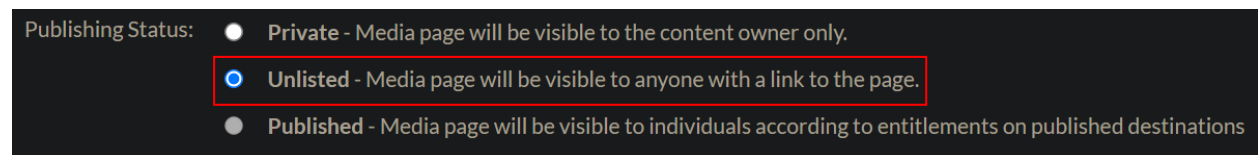
Your goal with your voice is to demonstrate to me that you understand what you did in this assignment. No voice track = no grade. Doesn't have to be radio quality scripted content here, just a quick demo like one you might send to your manager demonstrating a prototype.

**The video should be no longer than 15 minutes.**

You can use Kaltura to record your video. But if you use some other tool, that's fine too. Either way, **the end result must be uploaded to Kaltura.**

<https://docs.cci.drexel.edu/display/CD/Kaltura+Drexel+Streams+--+Adding+Media+to+Bb+Learn+--+Student+Instructions>

When uploading to Kaltura:



You must select **Unlisted** when publishing your video on Kaltura. If you select private, the grader will not be able to view the video – you will receive a 0. If you select published and another student finds your video and copies your code, you will both receive a 0 – so don't expose yourself to that risk!

Your final submission for this assignment:

- Link to Kaltura video
- Zip file including all code + UML (same instructions as the previous assignments in this course)
  - Zip file should have folders in it:
  - "server" folder with all Java files respective to the server code – including UML for just server code
  - "client" folder with all Java files respective to the server code – including UML for just client code
  - "common" folder with any Java files that are used by both
  - Do **not** include .class files

**No video = no grade. No voice track = no grade.**

-15% off if video is over 15 minutes long, even by a second. -20% off for each day late.