

CS-420-1: Object-Oriented Design
Fall 2019
Northeastern Illinois University
Homework #2: Due Thursday, 09/12/19 at 5:00 p.m.
Binary I/O

- Download the files provided for you from the course website and unzip them into a folder.
- Similar to Homework #1, import the `homework2.project` by doing the following:
 1. Open IntelliJ. Choose "Import project" and select the unzipped project file folder (the one that contains all of the files).
 2. Choose "Import project from external model", select Gradle and click Next.
 3. The Gradle project name/location should be filled in. Leave it as is. The checkbox "Create separate module per source set" should be checked and the radio button for "Use default Gradle wrapper (recommended)" should also be selected. Make sure that the Gradle JVM is using Java 11. If not, you will either need to select that version or navigate to the JDK and find it (see possible locations below):
Mac users: `/Library/Java/JavaVirtualMachines/jdk-11.0.4.jdk/Contents/Home`
Windows users (probably): `C:\Program Files\Java\jdk-11\bin`
 4. Click Finish.
 5. **NOTE:** All your code should be placed inside of the `homework2.project` → `src` → `main` → `java` folder. Do not modify any of the gradle files that have been provided.

The Problem

You have been provided with 3 files of differing sizes (these are located in `homework2.project` → `src` → `main` → `resources`: `test.txt`: 27 bytes, `alice29.txt`: 152 KB, and `largefile.dat`: 512.3 MB. The goal of this problem is to write a well-designed object-oriented program that completely reverses the content in each file. For example, the original content of `test.txt` can be viewed on the left and the reversed content on the right.

```
this is a sentence  
and more
```

```
erom dna  
ecnetnes a si siht
```

Your solution/code should adhere to the following requirements:

- Your solution should be organized in an object-oriented manner (i.e. all your code should not live inside the main method - make a class that can be used to make objects!).
- The name of the file to be reversed should be an input/parameter to the program (i.e. easily changeable from the main method).
- The name of the reversed file should be the name of the original file prefixed by the word `rev`. For example, the reversed file for `test.txt` should be `revtest.txt`.
- You should use a `RandomAccessFile` for some portion of the solution.

- You should take into account buffering in your solution.
- Your code should run in under **1 second** for the largest file.
- Make sure to handle exceptions with appropriate stack traces or error messages.
- Make sure all file resources are appropriately closed after being used.
- Print out the name of the file being reversed and the size prior to beginning the conversion.
- Print out the name of the output file and the time it took to complete and that the conversion was successful.
- You may not modify the gradle files or the provided data files (or the location of the files).
- Due to how Gradle handles resources, notice that when the source code is compiled, the resources are copied to the `homework2.project` → `build` → `resources` → `main` folder. This is also where your output files should be explicitly written to (help provided below).

Helpful hints/information:

- Use the very small `test.txt` file to start with - the larger file should be your last test case because it's binary and you won't be able to easily tell if it is reversed correctly.
- Important: Make sure you delete your reversed file prior to a new run that attempts to reverse that same file otherwise you may end up with more content in that file than you expected (and this could affect the time it takes to run!)
- Reminder: Your generated output files should go in: `homework2.project` → `build` → `resources` → `main`
- Do not hardcode the above path in your code. Instead, use the following:

```
String fullPath = <classNameHere>.class.getResource(<fileNameHere>).getFile();
```

 This uses the the current class that you are in to determine where the resource file is in the build directory and provides the full path to the file you are reversing (you need this path!!). You can also use this code to create a path for the output file.
- In order to time the file reversal, use the following:

```
Instant start = Instant.now();
// Your method call to be timed here
Instant stop = Instant.now();
long timeElapsed = Duration.between(start, stop).toMillis();
```

Don't forget to import these classes (Instant is a Java API class) - IntelliJ can help you with this!

Submitting your homework to D2L

- In IntelliJ, under File, choose "Export to Zip File".
Use the default name of `homework2.project.zip`.
- Submit this file to D2L.