**CS124 Code Generator Project – groups of 2, due April 22**

Given an existing generator project, you will need to analyse and reverse engineer how it works based on the input files in the /input folder and the generated files in the /output folder

In addition to submitting the code, you will need to defend this, i.e. you will need to explain to me why you did what you did.

**Requirements**

1. As in previous labs, you must apply the ideas of using the FastClasspath Scanner coupled with Interfaces, Annotations and Hashmap
   1. There are several places where these can be used
   2. HINT: look for the annoying if-else
2. MINIMIZE code duplication/cut and paste. For example:
   1. don’t have 2 annotations that are essentially for the same purpose, just share one
   2. if code in your processors have common code have a way to share this common code, e.g. using a base class via inheritance or other mechanism
3. Output must be IDENTICAL (i.e. field order must be preserved)
   1. Be careful about what you cut-and-paste or delete
   2. Always keep a clean copy of this project for reference
   3. A diff-ing program will be used to compare your output against the reference output

**Magis (for the big cheese, B+ up)**

1. Convert the Parser class to no longer use the single linedata format but Annotations from a model class (similar to our previous generators).
   1. WARNING: enumerating the fields/methods in an object is not guaranteed to appear in the same order as in the code
      1. The output is order specific, you will need to find a way to preserve order
      2. HINT: java.util.LinkedList and java.util.LinkedHashMap are your friends
   2. In addition to ordering, you will need to preserve grouping (e.g. in the single line format fields after a FRAGMENT are grouped until the next FRAGMENT appears)