

COS126: Final Project Report Template

Fall 2022

Instructions: Make a copy of this Google doc and respond to all required questions. Then, download this Google doc as a PDF and change its name to `report.pdf`. Finally, upload `report.pdf` to [TigerFile](#).

Note: Unlike the proposal template, this template is just a suggestion of the minimum information you should include. You can and should feel free to modify this template to suit the needs of your project, for example, by adding sections or questions.

1. Basic Information

Name #1: John Matters

NetID #1: jm2149

Name #2:

NetID #2:

Project Advisor: Gabriel Contreras

Link to YouTube video presentation: https://youtu.be/z_TiZaPKxH8

2. Project/Course Feedback

Before moving on, fill up the following form (<https://forms.gle/MSQC4cswCI3PBszY8>) to provide feedback on the project and on the course.

We filled up the feedback form: (x)

3. Project Description

What is your project title?

BachBot-SATB Chorales Generated with N-Gram Algorithm

Please summarize your project in 1 paragraph. Only describe things you implemented.

370 Bach chorales were parsed and put in a modified symbol table, based on user input the program then generates random chord progressions based on the symbol table. The result is

outputted through jMusic in both display and audio. I implemented a class to parse Kern files, a class to generate/analyze Markov chains, a class to store the Markov chain, and a client.

Please list all the relevant files (.java files, datasets, .jar libraries) from your project. Include a brief explanation of their contents.

Implemented classes:

BachBotClient.java – client

BachBot.java – Markov chain class

ChordST.java – contains Markov chain

Kern.java – parses Kern files

Libraries:

jMusic-1.6.4.jar

Chorales (“chor001.krn” – “chor371.krn”)

Describe instructions on how to compile and run your project (e.g. compilation commands).

In IntelliJ, first import libraries jmusic-1.6.4.jar, stdlib.jar, and introcs.jar (under “file” -> “project structure” for Mac)

Then, either build with IntelliJ (“Build” -> “Build Project”)

Or compile with LIFT in following order: Kern.java, ChordST.java, BachBot.java, and BachBotClient.java

Then run BachBotClient with appropriate arguments.

4. Features and Project Requirements

For each of the following, mention the .java files, classes, and methods that implement the things you are describing. If you used any external libraries, mention those as well.

Feature #1:

In 2-3 sentences, please describe your feature. Only describe things you implemented.

Kern file (.krn) parser. For feature 1, I implemented a class that can parse Kern files encoded in the Humdrum format from Ohio State University. The parser takes in kern files and is able to return arrays of parsed pitches and time values, as well as a jMusic Score file.

<https://www.humdrum.org/Humdrum/representations/kern.html#Pitch>

<https://github.com/craigsapp/bach-370-chorales>

In 1-2 sentences, please describe how you tested your feature.

A sample chorale is loaded in the main function of Kern.java, and StdOut prints various values.

What was the agreed category of your feature using the buckets paradigm?

☐ Standard

☐ Sprinkle

☒ Sparkle

Feature #2:

In 2-3 sentences, please describe your feature. Only describe things you implemented.

Modified Symbol Table and Markov chain for chord generation. ChordST.java and BachBot.java are the two implemented classes. ChordST modifies the ST data type by allowing int[] and int[][] inputs, which were needed to encode chord K-Grams. BachBot follows a similar model as the text generator assignments with detail improvements.

In 1-2 sentences, please describe how you tested your feature.

Sample values were loaded into ChordST and the output was printed to StdOut. BachBot was tested on a select few chorales to verify the outputs.

What was the agreed category of your feature using the buckets paradigm?

☐ Standard

☒ Sprinkle

☐ Sparkle

Feature #3:

In 2-3 sentences, please describe your feature. Only describe things you implemented.

Audio and visual output using jMusic library. Kern files and the generated chord progressions were displayed via the jMusic external library.

<https://explodingart.com/jmusic/jmDocumentation/index.html>

In 1-2 sentences, please describe how you tested your feature.

Played notes to ensure the midi is working as intended. The scores are displayed.

What was the agreed category of your feature using the buckets paradigm?

☐ Standard

☐ Sprinkle

(X) Sparkle

Other Project Information:

What would you try to add to your project if you had more time to implement extra features?
Instead of a markov chain, I could have used a recurring neural network algorithm in order to recreate Bach chorales with more accuracy. Also, identification of keys and modulations would help make it better.

What did you learn by working on this project?
Working with external libraries, practice working with abstract data types.

What advice would you give to students next year, to help them have a better experience working on their projects?
Start with a modest scope, and then reach towards higher goals

5. Extra credit

You should use this section to mention any of the extra credit items you did. Please provide justification for each one (for example, for the version control item you should have a link to your project on a website like GitHub)