**Lab2 - BellChoir - Documentation**

**Description:** This is a multi-threaded program that will consist of three primary classes: BellNote - Class to represent a note and its associated length. Player - Class that represents a unique player for a musical note. Each Player runs in its own thread and is responsible for playing one specific note. Conductor - Class is responsible for loading and playing a song composed of musical notes. It reads notes from a file, delegates playback to Note-specific Player objects, and coordinates synchronized audio playback. When ran it will effectively play a song given through a text file. Given a build.xml file, you will be able to run the program from the command line with string arguments. By default, the program will run Marry Has a Little Lamb, but in addition, I provided a file called AllStar.txt, which will play the famous Shrek song.

**Challenges:** By far the most difficult thing within this Lab was the use of syncronization tools in the player class. Initially, I had just the hash map that made a key a unique note paired with a player thread. It would look through the key and basically just call for the player to play the note, which did work and kept the threads separate, but I don't believe it met the requirements of multi-threading techniques such as wait(), notify(), and synchronize. As a result, I had a solution, but it had to be refactored to include these techniques, which took time. Apart from that, adding notes was a bit confusing, but by the end of it, I effectively got every note that you would need, apart from flats.