Abstract.
My goal is to create a learning tool for music students to develop absolute pitch. The project will consist of three parts: (1) informal and formal research in music and other areas to devise effective listening exercises, (2) algorithms development to devise effective methods of seeking patterns among the responses to these exercises to help correct a user's personal tendencies, and (3) software that encompasses all of this in a user-friendly interface.

Introduction.
Background.
Absolute pitch, also commonly known as perfect pitch, is the skill in which a person can recognize the pitch of a note without any context. (The ability to name pitches within the context of other notes is a separate skill known as relative pitch. Relative pitch will not be covered in this project, since effective methods of improving it are already known.) Absolute pitch is a rare skill, even among professional musicians.

A trained musician will usually demonstrate their absolute pitch by giving the name of the note, such as C or A#, but it is not necessary for someone to know the note names to have absolute pitch. For example, someone with no musical background could recognize a stand-alone pitch as being the same as the starting note of a specific recording of a song or piece, which could then be corroborated. In other words, despite its rarity, absolute pitch is a skill that may exist in anyone, regardless of musical background.

In the literature, it is considered impossible to develop absolute pitch after roughly age 9, and it is still highly contended whether or not everyone is capable of developing absolute pitch at all. However, some facts suggest that it may be a more accessible skill than once thought. Firstly, absolute pitch is more common among musicians (particularly ones who began young) and speakers of tonal languages such as Chinese, which suggests that the ability is developed at a young age through exposure and practice, rather than being a rare innate gift. Secondly, even among the general population, people are often able to sing a famous tune on its designated starting note, or within two semi-tones of it, with some 40% accuracy (source). To be fair, this is a generous margin of allowable error, but it is impressive that the average person can recall close to an absolute pitch, presumably without having ever consciously tried to memorize the starting note to any tune. Finally, many people without absolute pitch are able to hear a note and then sing back that note on the first try. This suggests that there is something in our subconscious minds that recognizes that note, as distinct from all others, that causes our vocal cords to constrict in precisely the right way to produce that note. At least for some instant, between hearing the note and beginning to sing it, the mind has absorbed and recognized that pitch, without needing the context of having sung or heard other notes. Though it has not been confirmed by research, I believe there is a reasonable possibility of coaxing the ear into developing these primitive beginnings of pitch sensitivity into fully-fledged absolute pitch.

Motivation.
Absolute pitch is not necessary for musicality – indeed, as stated, a huge number of very accomplished professional musicians do not have absolute pitch. So, why is it the goal of this project? Simply put, it is a personal hope to prove that something considered impossible is in fact possible. On the whole, I believe we as a society are often readily willing to give up on something that is deemed impossible, particularly if there is some scientific (or scientific-seeming) research to back it up. Absolute pitch is just one small example of a supposed limitation that I feel has not received either its due effort in attempts to develop it, nor in demonstratively showing why it should be impossible. Of course, I do not know if it is possible or not, nor if I will stumble upon a useful way of training it if it is, but I hope that
regardless of its efficacy, this project will provide some insights on absolute pitch, as well as general methods of training a rote skill.

**Approach.**
The philosophy behind my approach will be to step away from a black-and-white view of someone either having absolute pitch or not, and move more toward the view that aural recall of a note exists in different degrees in each person, and that it is an incrementally improvable skill. The idea is to create exercises of varying difficulties that gradually stretch one's capacity for recalling a given note, through time and through hearing other notes.

It is a simple exercise to show that someone can recall a given note, say A. One hits the key on the piano, or uses a tuning fork, and the subject hears the note. Asked to sing it immediately, the subject can do so. (We will assume here that she can, in fact, do this; people who cannot will be out of the scope of this project.) Asked to wait ten seconds, perhaps if she keeps humming the note silently to herself in her head, she can probably still do so, with essentially no dip in accuracy. How about a minute later? Her recollection of the note may start to drift a little, maybe by a quarter or a semi-tone, but if she is still mentally holding on to that note through the entire time, it is unlikely she will have wandered off much farther than that. However, after a certain amount of time it becomes entirely unrealistic for the subject to really be continually focused on imagining this silent note. Furthermore, this process of forgetting happens much more rapidly with other added variables, for example if other notes – particularly, random, chromatic notes that do not provide a recognizable framework with which to situate the note – are played in the interim.

**Customizable exercises.**
This suggests that a set of exercises, ranging among various parameters such as time and interfering notes, will allow the user to find the limit of their current recall ability and, through practice, extend it. The software will allow the user to choose settings for each individual parameter to increase the difficulty of the initially trivial exercise of singing back a heard note. In various combinations, these calibrated exercises will aim to pinpoint the user's specific capabilities and weaknesses, and work to improve them.

**Parameters.**
This list will likely change as the project continues.
- Note duration: How long each note is played. The longer it is, presumably the easier it is to try to memorize the note. However, making it too long may be detrimental as other notes are played, since it requires the user to memorize that note for longer.
- Number of notes played
- Which note of the string of notes to memorize (1\textsuperscript{st}, 2\textsuperscript{nd}, etc.)
- Range of notes played (how wide is the range, and also where is it placed – high, medium, low?). Note that a thinner range may actually in some cases be harder than a wide one, since the string of notes are more likely to be chromatic.
- Timbre of instrument
- Varying rhythm between notes
- How many exercises to perform in one set
- Whether you are memorizing a new note each exercise, or trying to hold the same note throughout many exercises. (Each difficult in its own way – speed vs longevity of memorization)
- Recognizing KEYS not simply notes? One might be easier?
- Only needing to recognize if the note appeared again in the string, and not having to sing it back.
Algorithms for evaluating data.
The software will then store information about the exercises the user has trouble with (and also the exercises he finds easy) to figure out what sorts of habitual mistakes he makes. It will compile this information into useable feedback, such as, “You seem to have trouble memorizing pitches below C3,” or “You seem to have trouble with recalling a note when the note directly following it is a semi-tone above or below it,” and then being able to provide a personalized drill in which this specific issue is practiced.

Potential supplementary projects.
In the interest of time, it is unlikely that my project will include either of these, but in the vein of developing absolute pitch, one side piece I would eventually like to add is the ability to load in a music file (e.g. a midi, mp3, or video file) and visually indicate every time a given pitch is played. The idea is that we must break the focus on relative pitch, which is essentially what we have been trained to pay attention to our entire lives so that we can recognize a tune no matter its starting pitch, and instead go back to an arguably much more rudimentary, yet now lost, skill of recognizing a given frequency.

A second piece is helping people recognize the starting note of various songs. This would ideally be fitted with similar feedback capabilities as the main project, which would give the user information such as the fact that they tend to misremember songs that start with D as always a few steps higher than they are, or so on.

Expected challenges and relevant resources.
I intend to read accounts of and perhaps interview people who have studied music, both those who began early and those who began late, to get a sense of how they developed their ear. Some informal conversations have proved somewhat interesting thus far, such as one piano player who recalls being able to recognize the pitches of white keys before black keys, and a cello player who can sing the tuning fork's A (440 Hz) on command, but otherwise has no perfect pitch. These accounts suggest that certain pitches can be ingrained into the memory even while a fully-fledged absolute pitch does not exist.

I would also like to do some basic research on existing theories of hearing, via biology and psychology, to see if these may shed any more light on potential ways to develop this sensitivity.

I also intend to look into machine learning algorithms to provide some way for the software to detect patterns in what sorts of exercises the user fails to get right. The user may be asked to provide some extra input, such as (if they got it right) how difficult it was to answer the question, or what their second guess for the note was. Coming up with an appropriate learning algorithm for the program to direct the user's next challenges will likely be the most intellectually interesting part of the project.

Finally, I will need to do a lot of work in learning how to program this software. I have essentially no background in making GUls (not required for any CS core classes, and only used minimally in any of my electives). The technical aspect of putting together a relatively large program and making sure it is free of bugs will be a challenge for me. As a part of this, there will be many design questions along the way, and in particular with finding an intuitive way for mistakes to be displayed and compiled into useful information for the user. My starting point for potential designs will be the existing software online, including sites such as teoria.com, which includes a range of exercises from recognizing pitches to harmonic progressions to seventh chords, and hooktheory.com, which focuses solely on harmonic progressions but may still provide some useful ideas for design. In particular, despite the somewhat technical nature of harmonic progressions, which often requires some music theory background to...
understand, the site's designed has managed to make its program quite accessible to users. For example, it does not even require one to know how to read music to follow along with the melody, and instead uses a color-coded contour shape to lead the listener along. Staying away from the technicalities of music reading and music theory, at least as an option (more advanced users would ideally be able to elect a more formal layout), is useful for at least two reasons. First of all, it of course increases the range of potential users, and second of all, it also helps divorce the experience of really listening to a pitch, which is what we want to emphasize, from the perhaps spurious associations that one may have been taught with various pitches or keys.