[SQUEAKING]

[RUSTLING]

[CLICKING]

MICHAEL CUTHBERT: You've seen that there's more technical problems than a standard class, and that means you're in the right place if you're here for a computational class, right? This is 21M.383 Computational Music Theory and Analysis. Or I can't remember if I call it analysis and theory. Don't hold it against me. I'm Michael Cuthbert. I'm the professor of the class. I'm a professor of music in music and theater arts, and I do some work with computational analysis. So what I want to start with-- oh, and that's who I am. I'm going to get to know who you guys are on Friday and next Wednesday.

It's very important to me. I want to really quickly be able to learn everybody by name. But on the first class, I think you're all here to figure out if this is the place you want to be before we get to know everybody else. So that's why I'm not going to be going around and asking everybody today, but I'm going to-- I really want to know that soon. So the big question for this class the whole semester is going to be, how do computers hear music? And what does that even mean? I'm going to give you an answer, if you care about answers like this, that you can read, absorb, pack up, and go home, and you've no need for this class if this is satisfactory.

Computers hear music by using a microphone connected to an analog to digital converter, connected to a microprocessor with a specialized codec-- I can never remember what that really means-- to quantize the sound into a WAV file, or AAC or something else, file that we store on a magnetic or solid state disk, or we store it on the cloud and we store it on somebody else's magnetic solid state disks. Not a very, very useful answer. So what I think we more want to know is, the computers are hearing music, but are they actually listening?

And these two terms, hear versus listen, in different contexts, one of them has higher priority than the other and implies a more deep, deep understanding, like mom or dad ever tell you, I know you heard me, but did you listen to me. We have that kind of distinction built in. So what we want to know is, when computers hear music, what kinds of understanding might they have, might be encoded in that, If. You feel uncomfortable with the notion of computers having their own autonomy or something like that?

And the only reason I really care about this question is that, by understanding how a computer hears, it might change how we hear, or it might open up new pieces for us to listen to that we didn't already know. And most importantly, it might answer questions about what it's like for us to be humans. Any androids here? OK. What it's like for us to be humans who are listening to music. So here are any six big unknowns for me that I want to know the answer to. I'll just say, you turn in on Friday the correct answer to any of these six questions, you get an A plus and you don't have to take the class. But I don't know them.

So why do we listen to music? Why? We don't go around all day. I have my different scents that I smell all day or something. There are all kinds of things that are enjoyable but that seem to be different, that we don't do all day long nowadays. Why do people choose what musical styles they like? How does your musical tastes, or our whole culture's musical tastes change over time? And some of us will be from different cultures and have different musical tastes. What's the difference between a great piece of music and a mediocre or bad one? Or do those terms even really exist?

What makes a piece of music beautiful or ugly? Let's not forget about the physical. Why do I want to dance or move when I hear some music, and other ones, I just want to sit and cry? So these are the questions that if you answer them, that would just be astounding. I think, however, that a one semester, twice a week class, we're not going to get the definitive answer to them. So I would like to break these questions into slightly smaller unknowns that help to answer these questions, and that we might be able to start making progress on.

What particular musical characteristics distinguish a folk song from one part of the world from one in another? How does the look of a band, its costume, age, its racial composition, affect the makeup of the audience? You know I'm not about to be a heavy metal star the way I'm currently dressed, right? Well, what does that mean? Maybe I'm OK at the piano. I'm actually better at metal than the piano, as you'll hear. But that's OK. I'm pretty bad at both.

What sorts of chords are used for relaxation, for rest, for stop in music today? And how is this different in different centuries or different parts of the world? And what do the lyrics of top selling songs have in common with those of worst selling songs, or don't have in common? And that's a question that I know that there are people at the big record labels trying to answer that, because there's money involved. What rhythms or instruments accompany texts about sadness or death that appear less often in moments of happiness? That's kind of related to the why some music make me cry and so on.

So five good questions that might help us answer a big unknown. What I'm going to do is I'm going to leave three up and take two off, and say that these three questions in particular, musical characteristics distinguishing x from y, chords used at certain points and change over time, rhythms about, and connections to emotion, these all might be questions that having a representation of a musical score, whether it's more modern, whether handwritten, ancient. These are questions that might help answer. So let's just kind of recap. What are some of the things the computer is good at with music?

It's kind of old slides. You'll see it's pretty out. But that we make a lot of music with a computer. We mix it. We do re-editing with it. Computers have absolutely changed how we distribute music, how we get it, how we access it. And then very cool apps and algorithms developed at places like this can tell what song particularly is some drunken jerk singing in a bar. But until about the last decade and a half, little had changed since the '70s and '80s in how computers could be used to analyze scores. And so things are changing now.

But let me say, where things were, what people already had for, let's say, text, that's different from music. So if you load a text-- we're all computer geeks, at least a little bit. I consider that as a term of endearment for myself, but I hope that's OK. We're all computer people. And so we can look at-- I can use the term string, right? We can look at a string of text like this, and you can ask various algorithms that already exist in any programming language to be able to identify the spaces, split, or what's a foreign diacritic. Cast it to ASCII and see what crashes. What's punctuation?

And then if you have some kind of font rendering, you'll know what's going below the line, and so on, what's a capital letter, all these things. And this level of low level information about text really helped to build the next level of information about text sentence diagramming, understanding from letters to a sentence level understanding. And this kind of built up knowledge has helped to make to the next level, which we might be arguing today about whether or not there is an actual meaning involved. I'm hungry. I want spam.

So can we build these technologies for musical scores? Maybe. Maybe, but not yet today. So this is a class that I hope people 5, 6, 7, 10 years from now-- hopefully less. Hopefully, not 30, 50-- that what you do in this class will help get to that level of understanding. I think put it somewhere. This is a unique, really unique class. I think that there's about three other classes that are something like computational music theory offered in the world today. The other ones are all doctoral seminars or graduate seminars.

This is a class that, because of MIT's unique ability to merge the arts and programming, a deep knowledge of music, and a great culture of collaboration in solving problems, we can do this at the undergraduate level. And if you're a graduate student, we can do even more with you all on that. So what are some of the technologies we need to build? Well, here's some simple things that a computer has to know to do anything with scores. Start off with D, E, F, G. What's the next note?

AUDIENCE: A.

MICHAEL CUTHBERT: A. Not H, right? Not even the Germans who have H would go on from that. So you need to know about these cyclic little things. Take a look at this next slide. Yeah, we got some squirmers in the room. That 3/4, this does not compute. So we need to have an idea of meter and things. And then from there, building up which of these two scores is more consonant and which is more dissonant.

[MUSIC PLAYING]

So we need a concept of consonance, dissonance, and how to bring this to the score. So we're going to need to step back from some of even the middle sized questions and go to even smaller things. So for instance, let's say this is the Tchaikovsky's 1812 overture. And let's see if we wanted to understand how the key relations work, how we move from one key to another, and how this builds up the form, and the tension, and the power of the piece.

Well, in order to do this, we need to know where all the key changes are. So we have to be able to cut the score into smaller pieces and identify that. Now, how might you identify a key change? Well, you might say, well it starts off with a cadence of on key, and then it ends in a cadence in another key. So now we need to be able to identify cadences.

So we need something that can do that. Well, this chord moves in. Five moves to one, so that's a particular type of cadence. There's that term. When I say five moves to one, are people hearing Roman numerals in your head and visualizing it? So these are the levels of musical knowledge that we're going to be using a lot here.

And so in order to figure out if a chord is a five chord, or one chord, or what's in the chord at all, if it's major or minor, we need a representation of chords, and notes and pitches. So some of what we're going to be doing at the beginning of the class is recreating some of these tools, and then we'll see what other people have done and go further from that.

So the notion that we don't have all these things is kind of surprising, because there's a lot of money, a lot of work that's been out there, taking some of representation of musical score, whether notation, or MIDI, or something like that, some of a score, and playing it really nicely on speakers and things like that, Garage Band, all those things.

Or if you've used Noteflight, Finale, Sibelius, Dorico, something like that, Flat IO, you know that we can take a representation of notes and turn them into some of musical score. So I think of that as kind of output. It's like a printer. And then a lot of work, especially in research labs, on taking sound in and converting it into that kind of digital representation, that's useful for search things. But once you have a digital representation of a score already in your computer, what do you do with it? How do you analyze it? How do you work with it? And there's been a lot less research on this.

So it's like we've been spending all our time building printers and-- I don't know how do we get things in-- scanners, and much, much less time building our actual CPUs. Nobody has a CPU that looks like that anymore. Time to update your slides, prof.

So one of the things that we're going to be using a lot in this class is the toolkit that I and many other people at MIT have used to do these kind of manipulations. And I think it's correct to say it's the most used toolkit in the world for working with musical scores. It's Python, free and open source. And sometime over the next couple of days, you're going to be installing it with Pip Install.

So Music21 really works from the notion that, first, we're going to encode a lot of ground up concepts about music so that when you have a chord-- is that going? Yep-- we'll just generate a chord, obviously. Not obviously. Otherwise, you wouldn't need this class. But you're going to have a representation of how to represent pitches, middle C, E, G, and then all the different things once you have that chord that you might be able to do, you might get its lyrics out, and it's going by really fast. Get its volume. Is it a diminished triad? So you can do that.

And hopefully, any good computer programs seeing C, E, G for is diminished triad will return--

AUDIENCE: No.

MICHAEL CUTHBERT: No, false, or anything like that. Zero. Yep, good. So in the amount of time it takes to write computer program that can tell whether C, E, G is a diminished triad, you all can hear it instantly. So why have a computer do things? And it's really so that we can listen to music faster than we can ever listen with our ears.

So this is a particular example of code in Music21, similar code in other toolkits, that will take a chorale by Johann Sebastian Bach, his 11th cantata, sixth movement, encoded as a file called MusicXML, store it as a variable, get the alto part out of it, store it as alto, get measures 13 to 20 as excerpt, do a particular analysis where we're going to put all these stars on according to how strongly accented something might be metrically, and show it.

And really, any music analyst, in the amount of time I took to do this, could have done it herself, or himself, or themselves. But what's the great three-letter-- is it preposition? Preposition that makes computers really powerful? Begins with F.

AUDIENCE: For?

MICHAEL CUTHBERT: For, yeah. So we just put a for loop in, and suddenly, we're looking at every chorale that Bach wrote, getting its alto part, measure 13 through 20, doing the same thing. And of course, why just do that little excerpt? Why just do the alto part? Let's grab every chorale. For loops within for loops. Don't over optimize first. Figure out if you can do this in the time it takes you to get a cup of coffee. Yeah, we'll come back to that theme in the class a lot.

In your computer science intro classes, you're going to early on be thinking a lot about, is this big O of n squared? Is this n cubed? Is this n log n, all these terms? We're going to talk about that from time to time here. But for the first part, we're going to build tools that work in an acceptable, to us, amount of time. You're going to hear me use this term a lot when we have a really big research question about, oh, I don't know, 15th century musical styles. And we'll say people have waited 500 years for this answer. 10 minutes isn't so bad.

And then later, you can go back and make it a three-minute problem, something like that. Some of the questions we might ask that come up that these kinds of tool kits are built to solve are higher notes louder. Many of us perceive them as louder. Are they given louder dynamics? So we can plot pitch. If we can go octave numbers, middle C, C4 is the standard American representation today. And the cello doesn't really go up that high. I don't know. Is Sforzando really somewhere between Forte? Who knows? These are kind of slightly arbitrary. But doesn't look like it to me.

Now, this is one little graph. Probably want to do this on a couple thousand pieces, run some regression lines, all the things we can do. But we can start answering those questions. How does Robert Schumann use notes differently from Frederic Chopin? And I plotted this in, plotting the pitch usage of Schumann. I can see how beautiful. It's a really nice bell curve. Really great. Chopin, on the other hand, looks, I don't know, kind of like one of these cities with no planning laws, let's say, Abu Dhabi with a gigantic skyscraper.

When Chopin likes a G-sharp, it's not just any G-sharp. It's this G-sharp. They just keep playing it forever. So there are some of the ways you can do this. Any Robert Schumann experts out there? No. Some hands almost went up. Some people like Robert Schumann. Garbage in, garbage out. This particular piece by Schumann is a string quartet. This piece by Chopin is a piano. When you have a string quartet, of course, you're going to have more instruments with notes in the middle.

So I did this once. I thought I had compared apples to apples, and I realized I compared apples to oranges. So we're going to think a lot about user error, also, in how we make things. Visualization of music is a big part of what we're doing. These kinds of visualized graphs can make understanding a piece very easy. You can also use 3D graphs, which don't make it easy at all to understand anything, but they look really cool if you need to donors, and funding, and stuff. So always make them when you need that. But otherwise, I can't really read that.

We can combine different fields, so for instance, geographic information and plotting, with musical characteristics. So this is one of my fun things I found. Comparing the distribution of large intervals, these are intervals close to what would be Western sevenths. But obviously, they're not exactly that because different cultures use different intervals. But looking at the folk songs differing, and you can see a large difference as we move northward, and a little bit as you move inland in China, in the increase of large intervals that are not octaves in folk songs.

And we'll look, at a certain point over the semester, at a very cool paper that kind of shows how particular mountain range divides folk songs that are very different on either side of that range. If you're looking for particular fingerprints that might tell you who a composer was, maybe there's these small moments you might find. There's one moment in this piece from around 1500 that feels like it might be in a modern composition, where you just smash your hand down on a piano.

It's very hard to see here, but with a computer, what you can do is we can take all of the notes and put them into singular chords here, reduce them all so that they're in closed position. Everyone remember that from music theory class? As close together as you can make it. Annotate the intervals, and suddenly, this one little moment jumps out as very strange. Then you can go back and figure out, how did the music, create this?

Over the course of the semester, we'll be looking at changes in music over time. And we're also going to be looking at various things that were done by people, either in this class, or who had worked in the lab that helps create Music21. So Beth Hadley was somebody who graduated from this program, who did this analysis. And looking at a lot of rock popular music songs in the 1950s and '60s, see the bass moves quite often by five semitones up or down, which corresponds to a movement of one to four, or five to one, things like that, and then occasionally, up by two semitones.

What's that in the 1, 4, 5, 1 progression? 1, 4, moving to 5, to 1. And so you can see how this really dominates. Did that go?

AUDIENCE: Yeah.

MICHAEL CUTHBERT: At the time, most data set, it says post 1980. It should say 1980 to 2000 data set. And you can see, if we go back and forth, how it's still these fives and twos that the main thing. But these other intervals are continuing to move up and becoming more and more important. Love to see an analysis. We now have another full 20 years, 22.25 years by the time the semester is done, worth of music after 2000 to see if this trend continues or stopped.

Anybody, y'all have taken either some of a theory class, so at some point. Who has written a parallel fifth in this class, accidentally? Yes. Wow, some people have never written a parallel fifth. That's amazing. Or you always did it on purpose just to annoy your teacher. So next time you. You've given that you can generate your own list of every single time that Johann Sebastian Bach wrote a parallel fifth or an octave, so that you can give it to your teacher and say, yeah, that was a parallel fifth. But it's exactly like what Bach wrote.

Once you've written a program that does a particular type of analysis on a particular corpus of music, a collection of music, an analysis, it's easy to swap one or the other, so that once you have something that finds all parallel fifths in Johann Sebastian Bach, then we can put it into Claudio Monteverdi, and find the same analysis and do different things. So each of our tools can work on-- what is it? Most of our tools-- I'll say not all of them. Most of our tools can work on any corpus, and any corpus can apply to multiple tools.

So this is a place where you get a squared increase in power by using the computer over if you wanted to do this by hand. I'm trained primarily as a musicologist. That is a music historian somebody first trained to do things by hand, and look at history, and look at context, and come to conclusions. Then I did a lot of computer things. And for a little while, these were slightly separate worlds. I'm trying to move them more and more together, so that when we answer a particular question, we think about which of those big questions, what bigger answer might we have come to.

So this was a particular demonstration program I wrote early in my development that I was like, oh, cool. You're going to see. Well, we'll talk about why JS Bach is coming up a lot. But I have all these pieces by Bach. And I know that Bach, when he writes a piece that's in minor--

[MUSICAL NOTES]

How is it going to end generally? Anybody know?

AUDIENCE: And on a C major chord.

[MUSICAL NOTES]

MICHAEL CUTHBERT: Yeah, I call it Picardy third. So usually, when he's in minor, he ends in major. I wanted to find every exception to that in his choral composition. Great. So pretty easy to find all these places. And then I was pretty happy that I found that there's exactly eight of them. I thought that was kind of neat, until one day, I was a little bit bored and I started looking at what these pieces were, what pieces went with these weird numbers. And I looked at the first one, piece 248 in Bach's catalog. And I saw, oh, that one's wrong. The computer didn't get it. I didn't expand all the repeats.

So the final bar in the piece then said, dot capo. Wait, which way is right for you? Yeah, da capo al fine. So go back there, and the actual end was in the middle, and that one was in major. So it followed that. Then I looked at these other pieces, and I saw they were what some people call orphan corrals. They're ones that JS Bach's son, CP Bach, said dad wrote and published them, but we have no other evidence that dad wrote them. And the only time this ever happens is in those collections.

So some people have been arguing, maybe these pieces aren't really by Bach, or they've been edited heavily. This seems to give us some of the evidence that might be true. So always remember that the computer is not substituting for analysis. It's a tool to make the kind of analytical questions you might already have more efficient. That's why. Yeah. That's what we know.

The first couple of things that we're going to be looking at in this class are how we represent music, what kinds of scores are happening, and so on. And we'll get into why you might do one representation or another. One of the first things we'll look at is music as a series of boxes, of Russian nesting dolls, whatever you want to call it. Box, within a box, within a box. So we have a score. Inside that box, we have some parts. Inside those boxes, we might have measures. We might have voices, two different independent voices within a measure, something like that. So we'll call those part based containers. You might even put the score into a bigger box with the Opus or all the works in a collection.

And then our second box we might have is everything that happens at a certain point in time. So everything that happens here, everything that happens here, here. And we'll be looking at and trying to identify when one representation might be better and one, another representation might be better, and how you might use right tools to move from one representation to another, from a box based to a moment based. You'll see we sometimes call this salami slicing. You just chop the thing like a salami.

A really important moment, and I alluded to with Bach there, is to try to make sure that the conclusions that we're making apply to more music that's out there. And that's only possible if we have a lot of music, and that music is representative of people who create music.

Here's some of my favorite composers out there. We have Ludwig van Beethoven. We have Joseph Bologne, Chevalier de Saint-Georges, Kate Soper, friend of this music department, Amy Beach, Wagner, Richard Wagner, JS Bach, Scott Joplin, Queen Lili'uokalani. I spend part of my year in Hawaii. Very important composer. Let me see. Yamandu Costa, Wolfgang Mozart. We've got Dolly Parton and the Beatles.

So these are people that I would really love to do my work on. So I go and I start doing the work. And of course, I don't want to encode all of this music in, so I go with what other people have encoded. And that lets me work on this subset of music. Very representative subset of the people who were up there before?

No, not at all. We have Beethoven, Bach, Wagner, Mozart, all great composers, long dead white males, out of copyright. And then still in copyright are white male group, The Beatles, that somehow, there's just always Beatles everywhere. You can almost always get that.

So this was the state of music, which sometime is called corpus research, work on large collections of scores, a corpus body, a large body, a repertoire of scores when I started going into this. Thankfully, the situation is getting a little, little better. We now have the complete piano rags of Scott Joplin, who wrote The Entertainer, Maple Leaf Rag. If you don't think you know his name, play some. We'll play some of his music, and you'll think, oh, yeah, I know that music. So very important composer we're starting to get in.

And individuals and groups are starting to create collections, like this collection of a couple hundred scores by women and nonwhite composers. I hope that this class, at some point, we will all contribute onto improving the situation just a bit and try to make it so, eventually, what is represented is in our representations looks like the world we want to be in.

So you have in front of you somewhere a syllabus, as well. So grab the syllabus. We're not going to go through it line by line. In fact, I haven't even given you yet the longer one that's going to tell you day by day what we're going to do. Because part of that depended on how many of y'all showed up today.

And it's a good, good crowd. Everybody comfortable? A little bit, a little bit not. It's over there, but we'll somehow make this happen. You'll be happy that this class, there's nothing to buy, unless you don't happen to have some of these things earlier. We'll use as a textbook the Music21, the standard textbook, the Music21 user's guide. Probably get to maybe Chapter 25. I always say 1 to 30, but we might not get there. And we can't go too much past 30, because it just kind of trails off. It's still being written. You can always write some more of it if you want as part of a project.

You're required for me to support you, for me to be a help to have a Windows or Mac OS laptop. We're going to install the latest version of Music21, which is, I believe, 8.1. The hope, the plan, the bags under my eyes right now was to get version 9 released before today. It's not out. It has some of the newer features that we'll be using. I'll probably get that out within a week or two. Now, you all are my focus, and then we'll put that on. So at some point, we'll all upgrade to that. Music21 version 8 runs on Python 3.8. Version 9 runs on 3.10, so install of Python 3.10. You'll be very happy with that or 3.11.

Probably want to get some staff paper and a pencil, because at various points, unless if you're using a tablet, you're probably fine. But it's sometimes hard if you're taking notes on laptop and we'll, at some point, be using that staff a lot. There's four recommended books. I love them, but they're all ebooks, but you might consider buying them. At least the first two are pretty cheap and things, but we don't really need to talk about that more. OK, we're making great time. The typical week, we're going to, on Wednesday, introduce a particularly new topic. This is just middle of the semester,

And then you're going to get some readings to do for it. Some of them will probably be for Friday. We'll try to make Wednesday to Friday kind of a time more for reading, and reflection and things. On Friday, so maybe the topic is-- oh, let me not be hypothetical. What is Topic 3? It's written there somewhere. Containers. We just talked about this, how to represent as containers. What are the various ways working through that? So we'll just introduce, like we did now, a new topic and think about it a bit.

On Friday, we'll get to some of the nitty gritty, difficult things, discussion of why this representation work, why this doesn't, why you might make this choice. And we'll do some in class work on that. And then usually, on Fridays, you'll get a problem set. So you'll get that on work on it. But over the weekend you'll also be getting a little bit of asynchronous content. So what happened was-- nobody really cares about the history of your class, but I'm going to tell it anyway-- we started this class. It was an hour and a half, twice a week.

And then COVID hit, and suddenly, a lot of things went online and I did a lot of video recording, partially because my wife and I were teaching at the exact same time in a one-bedroom house that had no way for us to both teach. So we did some asynchronous things and a lot of the asynchronous content I'm so glad you will never have to see again.

So you all lived through that. But some of it was actually better than in class, because sometimes, every once in a while, there's something that I can teach you a particular technique, and then if it was in class, I'd say, great, so now you know how to do that, and let's go on. And here's another technique, and let's go on. And you haven't had any chance to practice any of them.

So there's some things that are useful about the Sesame Street style of here's a particular thing that takes me about five minutes to present, and then you work on it for 10 minutes. Then here's another five minutes. You work on it 10. Before you know it, you've had one hour of lecture and you've had about three hours of a problem set that you've done problem by problem. So those ones are still there. There's about seven of times during the semester. And I just guarantee I wouldn't be doing that if I didn't think that was a better way to do things.

So you'll do some of that work there on Wednesday, and this will be at least four or five hours before class, so I can do a quick spot check-- that's the great part about doing things all online-- turning the problem set, and then we'll unlock the tools. What does that mean? It means if you're working on something, let's say we're doing something with cords.

I'm going to ask you to figure out certain of the algorithms for how to solve some difficult problems with chords. And most of the things that you're going to be doing, somebody in the world has solved. I hope that's not too disappointing. Most of what we do in most of our classes in structural engineering is saying somebody has solved it. We're learning to do it ourselves.

So at that point, we're going to show you some of the tools that are already out there to do this, and they will become unlocked to you. That is to say, you can keep doing, using the tools that you built, or you can use the standardized chord manipulation tools. And then we'll introduce the next topic and keep going over here. It does mean that you're sometimes going to have two topics going simultaneously. And I hope that's not too weird. I'm going to try to make that as unconfusing as possible.

So if you could get out your laptop and click on this button, I'll click on it with you. I'm just going to get this ready to go. When it all loads, go to Runtime, Run All, and then stop. That will probably take two to three minutes. OK, so we'll go on from that, and I'll just talk a little bit about final projects while that's going. There we are. Great.

So what we're going to do at the end of the semester is have final projects. We're a little bit big class. I'm probably going to let anybody who wants to have a group of three do that because of the number of people here. But generally, two works really, really well. Solo groups are really hard, and I found that groups bigger than three have almost always had some problems. So we're going to do a final project. And I used to say any topic in computational music or music theory and analysis, or composition type things, as long as it involves representations of scores for the computer.

I now have one exception. A lot of AI deep learning for composition projects have tended not to do as well as the students who ran them, were doing in class. So I'm going to not allow that for this semester to try to figure out how we can do that a little bit better. So that's just me learning from experience. You're going to spend some time researching a topic that you work on, programming the tools needed to do it, reading about what mistakes you might have done and just thinking about it.

Then what I want from you is, then you're going to be making a video about your project, five minutes. I know I'm a hard professor. I could have made it 10 minutes and it would have been a lot easier. But five minutes, you have to be on your toes. So what I want a couple of days before that is just a draft of your script. I don't care how messy it is. I just want to see that I'm not going to get a frantic phone call at 11:30 PM that none of our software works or something like that.

And then you record and edit a five minute video. So let's get back to that live code. And somewhere on here, this one, we're all up. And where do I have it? I have mine here. Let me make this bigger in three ways. Anybody not get a note here? And maybe this one sometimes doesn't work for a lot of people. So don't worry if that doesn't work for you. But give me a thumbs up. We got it? OK, cool. So let's start with me finding my script. Great. So let's do some of the things we can do.

If you don't care about polluting your code base-- so I got this a little bit-- go ahead and import from Music21. Import Star. It should already be in there. Has anybody not used Jupyter Notebook or anything like that? Who has used one? So don't need to do too much but we can-- yeah, we'll enter in a new code thing, move it down. So go ahead, enter a new code bank, and let's just make sure that if I type chord and hit Shift, Enter, something like that comes out. Cool. Great.

So there's a particular piece by Bach that we'll use quite often in this class. I'm going to use it less often this year because we have more alternatives now to this, and we don't want to perpetuate the "Bach is everything" stereotype. But this one we'll use a lot. So we'll assign to the variable Bach something from the Music21 corpus, calling the method parse on it. And we can just type BWV 66.6. I am not a devil worshipper or something. 66.6 happens to be the shortest chorale that Bach ever wrote, so it's a nice one for us to work with here. And Shift, Enter, and that should load.

If something doesn't work, this is the time to cheat and look up your neighbor and see if maybe you mistyped something or tap them on. We're all in class. We're all here always just to help each other, right? We feel that? You're going to do that for people around you. Good.

So let's show it. The first time you run show, it's always a lot slower than every other time. Hopefully, it won't take too long. You should get a score up there somewhere. Good. Right now, you're only going to get the first page of the score. So that's why I use this one, because it's only one page long. Soon you'll get the whole score.

So let's go ahead and let's get that alto part. So we'll say alto equals Bach. Parts is a way of getting something out alto, and then alto dot show. Everybody see something? It looks a bit like this. Alto part's always a little--

[MUSICAL NOTES]

[SINGING]

Something like that. Good. You can also get a part by doing it numerically. So anybody know in the course what are the four parts from highest to lowest?

AUDIENCE: Soprano.

MICHAEL CUTHBERT: Soprano?

AUDIENCE: Alto?

AUDIENCE: Alto?

MICHAEL CUTHBERT: Alto? Tenor? Louder, louder. Come on. Participation grade. Let's already get our thing. Soprano, alto, tenor, bass. Great. So if we can do it by numeric order, if we want to get the alto, that's what number part? Two, one. Yes, zero index Python. So what are the things you'll get, and so that should be the same part, parts one, good.

Good guesses. Good wrong guesses. I love great wrong guesses better than even getting it right on the first time. One of the things you'll find is, if something is in computer speak, we leave it alone. So we don't tend to number the parts in the orchestra. Hey, instrument 17, do this. No.

So when we do that, we leave it in computer speak and make it zero indexed. When we leave things that musicians might know, however, we tend to use the musicians thing. So let's get the soprano part, part zero here. And let's get measures four through six, so dot measures 4, comma 6 dot show. And in Python, that should get you the fifth measure through the sixth measure because of the way zero indexed. However, measure numbers. Measure numbers something that human musicians talk about? Yeah? Yeah. You've heard of that? Great.

We'll save time always by just being fearless at shouting out when I ask a rhetorical question. Measure numbers, we know, musicians? Yeah, good, good. That'll just save time in the class. So yeah, since musicians know what measure four is, we'll use measure four that way. So that's measure four through six. Great. So let's start manipulating the score. Let's say that the--

AUDIENCE: Four through six, or five through seven?

MICHAEL CUTHBERT: Thank you. That is whatever a musician thinks measure four is. So in this particular piece, let's go up to the beginning of the piece. Yeah. What measure is this that the mouse is on?

AUDIENCE: Zero.

AUDIENCE: Pickup.

MICHAEL CUTHBERT: Yeah, Pickup is what a musician would say. We got to give it a number. So what number do we like here? Zero. So there can be a measure zero if there's a pickup, but measure one will always be whatever humans, musicians think measure one should be. So we can play some other things. Let's just play around with the score a little bit. We'll take the soprano, get measure five. You'll also see some of our display things don't know how to display the right number. If you're on MuseScore four, it now displays the actual correct number.

But so it's measures four through five. Let's make the fermata red. So what number? If I'm getting notes, I'm going to get an index by number. So we're going to get measure five and get a particular number on that. What note number is the fermata? Shout it out.

AUDIENCE: Three, two?

AUDIENCE: Three.

MICHAEL CUTHBERT: Two. In this case, musicians don't tend to talk about-- I guess we do say the second note. Yeah, yeah, that's really confusing. So in this case, basically, when we're using the bracket, square bracket notation, I'll make this a little bit bigger. Does that help a bit? Yeah, we'll think of it-- computer. So we'll say C. I should call it C-sharp equals, but then you just get what comments for the rest of the line. So we'll just call it C, even though it's a C-sharp. C style-- I think I can just check. Nope. That doesn't give me auto-complete-- equals red. And soprano, show.

We got the soprano. I'll scroll back up again so you can see. We already got the soprano. We take a particular note, the third note indexed by 2 from measure five, which is actually 5, and assign it here. Probably, we could have made different decisions on how we number things if for Version 2, but whatever.

And then we'll set the style of that note's color to red, and then show the entire soprano line. And we get something over here. I'm half colorblind. Is there actually a red note up there? OK, good, good, good. Where is it? It's over here. We'll emphasize that for the camera. Shine a laser on it. It becomes more red. Now, yeah, that's kind of fun play-around things.

Some things we might actually want to do, let's take the whole piece. We called it Bach, right? Zero index,yeah. Bach's the score object. Let's analyze its key. Analyze is a method. So we call it like this, parentheses, parentheses, score key. So let me say, if you're following along at 80%, but you just wish I would go a little bit slower, and then you'd be able to have it, it's going to be fine, because most of the time, we're going to go slower than this. This is just the fun beginning of the semester.

If you've never seen a parentheses or square parentheses, this might be a class to take next year or next semester after doing a little bit more programming. And that's probably more for the camera, people watching at home. Let's make sure that we just need a little bit of programming before we're going to go forward. So the piece as a whole, F-sharp minor. What do we do now? I've tried to emphasize it early on, but we're going to make this a habit in the class. We always check our bs meter It says the piece is in F-sharp minor. Is that at all plausible looking at this piece?

Let's just look at the base. It's easier. Well, it looks like an A. It's the three sharps, maybe. Maybe. The end, F-sharp. A-sharp. Oh, it looks like it ends at F-sharp major. But then maybe the rest of the piece is in minor, and then this is just the Bach ending thing. So to me, it passes the smell test. It might be other things, but one of the other things we can do is let's analyze measures. Oops, we get Bach dot measures, 0. I haven't spoken out loud while I typed for a while. 0, comma 2. So we'll just get the pickup and the first two measures, and we'll analyze the key of this opening.

A major. That feels more right for the opening. So we know that the piece as a whole is in F-sharp minor, or at least the computer thinks so, if the computer got it right. And it opens an A major. A logical thing that we might play with next is, where does it change? Can we do that? Can we analyze where key changes are? We're not going to do that. Let's get some easy thing. So we said that not notes I used to get a particular note. So let's see how many notes there are in this piece. Len. Any other language would be dot length, but we can len Bach dot notes. And how many notes? Make a prediction? Zero.

There are no notes in this piece. We started with cage 433, and we end with Bach 433. Why are there no notes? It's because Bach is a score. This is all about the representation. We're currently in box, within box, within box There are no notes in the score. There are notes within parts. No, there are no notes within the parts. Parts have measures. There are notes in measures. So what we'll need to do is, we'll say, take the Bach, and let's run recurse. Go into every box like a child at Christmas. Open every single box and find every single note in there.

So we'll say we can take a score. We can say open up every box, and to every box, get me the notes. And 165. This passes the smell test. Let's scroll up. Don't ask anybody ever to count this, but that look like about 165 notes to you? Yeah. Just order of magnitude. If it said 1 million notes, would something be wrong? If it said zero notes, you knew it was something wrong. So we're always trying to do these kind of smell tests before we go on.

And the last thing I'm going to do, because I'm looking at the time-- and I have only one good habit as a professor. I really am allergic to keeping you over. So we're not going to do this, but we'll do just for fun Bach dot chordify dot show. Now we've switched the entire thing into salami slice mode. So these are some of the things we're going to be doing. Hopefully, that was good professor mode. Now here's really evil professor mode.

You're going to forget every tool that you've just been given. Analyze key is not available to you. I'll call it locking it. It's locked in this little cabinet here. When you write your own pretty decent key analysis algorithm, you can unlock this one. Yours might be better. I might ask for a license to, hey, can we use that one there. This is a pretty decent one, but you might write a better one. When you write your own chordification algorithm, then chordify will be opened for you. We'll give you some tools, extracting things, particular measures.

That was a lot of work, but that kind of sounds boring and boring in terms of our intellectual development. But coming up with the tools to be able to chordify a score, we'll see how many difficult things that come from that. And the time's about up, so we'll talk about that in other classes.

As I said, on Friday, I want to get to know you guys. I want you guys to be able to talk more. But I wanted today just to blab about what we might do in this semester. And I'm happy to have you here. The numbers look great. Y'all, come back Friday. Y'all, make sure that you're registered, and I'm glad to have everyone in this class.