[SQUEAKING]

[RUSTLING]

[CLICKING]

MICHAEL CUTHBERT: Hello, everybody, and congratulations on getting problem set seven in. This is Michael Cuthbert, and I'm going to unlock a new set of features on chords for you now that you have successfully implemented some of them yourself. So I'm going to go ahead and import the chord module from music21. And we'll remember that within the chord module is the capital C Chord class.

So we'll say C major equals chord.Chord. And we'll say C4, E4, G4. By the way, when creating a chord, we can create it a couple different ways. You can put it as a list of strings. We can make it a single space separated string. Or if we really want to do this, if we want to be painful for ourselves but sometimes that's how we'll do it, we'll create c4 equals pitch.Pitch C4, e4 is pitch.Pitch, E4.

The next one, g4 is pitch.Pitch G4. And then we can make C major equals. A list of those already created pitch objects. And depending on what context you're doing, one of these will be easier than others. We could also do it with a list of note objects, but I'm not going to demonstrate that now. In any case, we have C major and we have this chord going on and that's all nice and fun.

OK. I'm going to do that. So within this chord, one of the things we can do is we can get the root of it. And takes a little bit of time, so I've made it a function call-- a method call on that. So that seems to work. Get the base of the same chord. And that's the same thing here because the inversion is-- oops. The inversion is zero. Great.

Now let's do C major 6, not in the chord symbol sense, but in the Roman numeral figured base 6 as a first inversion. We'll create chord.Chord. Think through it before you copy down what I'm going to write. Great. E4, C5. Oh, let's go ahead and put an open position. C major 6 root. It's still C bass. It's still an E, but now-- oh, is now an E, because the inversion is 1.

Great. We can set the inversion here and see that everything has moved up above the G and moved all the way up. It's pretty far up there, so maybe we'll do 6.transpose inPlace. Oops. Transpose down. Let's say down two octaves. InPlace equals true, and now we're here. And I'm just realizing that this inversion thing should have returned a new chord or something like that.

So I'll put in TODO msc, make inversion not return-- not inPlace. And just in case you're watching this video later and your inversion does something else, I fixed that problem. OK. So these are some of the things that we can do here. Notice, let's go back to create our original first inversion chord.

Great. And some of the other things we can do. Those notes are pretty far apart. So we can say cmaj6closed equals C major 6 in closedPosition. And now we see C major 6 in closedPosition. Still keeps that E in the bass but figured out that, hey, that G could move all the way down to octave 4. Because C4 would be below E4, C5 only gets left there.

So these are some of the things that we can be doing with the kinds of things that we're doing with chords in our most recent problem set. Here's a little interesting thing with roots that I'd hoped to get to in class, but now we can talk about. It's one of these strange moments in music. So one of the things that came up in our discussion was, of course, finding the root.

And when you were working on finding the root on something like a C major triad, OK, that's pretty easy. Then you end up with a little bit harder is C major triad in open position. And maybe even harder is let's do it in open position and in inversion. In all of these, of course, the root is going to be C. But how did we do this?

I haven't had a chance to go through everybody's assignments yet, but there's different possibilities, some of which are more efficient, some of which are less efficient. But basically, whichever route you chose, it ends up usually trying to find some way to permute each of the nodes. So maybe we rewrite them all to be in the same octave, so tune, tune, tune.

And then we try various permutations. Maybe we rotate below. So we take the top note and move it down. So down to here. And we have a template in mind. And that template says that when you have stacked thirds, then the lowest note is the root. So here we don't have stacked thirds. What do we have? We have fourths and then a third above it. We would say 6/4 in figured bass, but think of it this way.

And here, we don't have it. We have a third and a fourth above it. We call that 6/3 position. And we don't have it, so we go one more time. And here, we have this, which we know is 3/3 or 5/3. And this is our perfect solution. And yay, we're super happy. And the cool thing is that this kind of work works just as well with seventh chords. So let's say we're doing-- here, we're going to use some of the same notes, but will it be a dominant seventh type position in F major, something like that.

And we can again start moving things into the same octave. I always make little noises when I make chords. And we can see, nope. What do we have here? We have third, third, second, so that doesn't work. We'll take the top note, bring it down an octave. Oops. Just anal enough that I want all my notes to be in blue today. So we can do something like this.

And voila, we have-- what do we have? Third, third, third. And we're all happy. And so the root is C again. But what happens if we have a ninth? Well, I won't make it a ninth chord. You can, again, put it into this particular position. And voila, we have C9, whatever inversion it's in, and so on.

But let's take a second and listen to a piece of music that's going to challenge some of our assumptions about what does a root mean, maybe go beyond what we've seen in our theory classes, but the type of things we have to consider as computational music analysts, whatever we are. The piece we're going to discuss is by Benjamin Britten. It's his opera "Peter Grimes" from 1945.

And we're going to be looking at one of the last scenes of the opera where the main character, Peter Grimes, has become deranged and lost his mind, and believes the accusations that have been made against him, that he has been killing his assistants, when, really, the most recent one has just died in a tragic mishap. And the music that we're going to be listening to, there's no singing. It's the music that accompanies him going out to sea to drown himself. So it's a very somber moment. I want you to listen particularly, though, to the arpeggiated chord played by the harp.

[BENJAMIN BRITTEN, "PETER GRIMES"]

Let's look at that chord from the harp in detail. It starts in the bass clef with F, A, C, continuing up E, G, B, D, and then coming back down. And when we put it into treble clef and stack everything up, we see that F, A, C, E, G, et cetera, we basically have a whole bunch of stacked thirds going all the way up. And so that's an F13 chord. So we know the root because we see all of the stacked thirds.

But now let's take the root and we'll move it to the top. So we'll start on A, C, E, G, B, D, F. And hey, that's also a stacked third. So our original chord could be an A13 chord in first inversion. I don't know what the figured bass symbol is for that. But then we can also take the A and put it to the top and see that, hey, C, E, G, B, D, F, A, that's also a stack of thirds. So this is a C13 chord in second inversion.

OK, let's see what happens in music21 when we try to do something with this. So we'll import everything. First we just need chord in this case. And let's create an F11 chord. And so we'll do it the same way as we're doing with the "Peter Grimes." So F3, A3, C4, E4, G4, B-flat 4.

We're leaving out that high D for now. And when we look at F11, its root is the F3. Its inversion is in root position. Great. Let's take that B-flat 4 down to the lowest part. I'll put it in B-flat 2. Now something like that. And when we find the root of this, it's still in F, and it's in the inversion where the 11th is in the base, which we all learned in-- no, it didn't. Is fifth. Oops. Is fifth inversion here.

OK. So that's going along pretty well. Now let's create the F13 chord in inversion. And we'll put same, that B-flat 2, F3, A3, C4, E4, G4, and we'll add that D5 at the top. And what used to happen when you called root in this point is it would crash until you couldn't find the root of this chord.

Eventually, I found that to be kind of annoying, that you had to code around a special case that you could have a 13th chord or something would crash. So I've just decided that the root of a 13th chord is always whatever note is in the bass. So in this case, now I put the F in the bass.

Now that's the root. And the inversion is always in root position, which is kind of-- that's also maybe a problematic thing that you can take the same notes, rearrange them and put them in a different place, and the inversion stays the same, but the root changes. So that's just a little thing on how 13th chords are a little bit different than every other chord, and why computational musicology and computational music theory is so much fun. Thanks, everybody.