

Lab: Band in a Box

1. In this lab, you will create a program that takes a simple musical melody and transforms it into a complete arrangement for a three-piece band. (Don't worry if you can't read music!)
2. Download band.cpp from the class website. Notice there are already two components written for you, a note struct and an instrument class.

Explanation of the note struct: A note is represented by a pitch: a character from 'a' to 'g', and an integer denoting an octave:



Explanation of the instrument class: This class has only one method, called play, which takes a vector of notes as an argument. The vector of notes represents the sequential notes of a melody (assume each note is an eighth note if that means something to you). This function is designed to take the melody, and depending on the instrument type, display an appropriate set of notes for that instrument to play.

Notice the strange prototype:

```
virtual void play(const vector<note> & notes) = 0;
```

The "= 0" at the end of the prototype means the play method in instrument is a *pure virtual function*. This means that in the instrument class, the play method is undefined --- it has no body. Notice the play method is never defined below the class at all. It also means that the play method must be overridden in any derived classes. Normally with class hierarchies, derived classes can choose whether or not to override methods from the base class. However, if we declare a method in the base class to be a pure virtual method, C++ will force all derived classes to override that method.

3. Write code for the play() method in the guitar class. The function should print the word "Guitar:", then "play" (by printing to the screen) every note of the melody.

Example of what would be printed:

Guitar:

```
c4 c4 d4 e4 c4 e4 d4 g3 c4 c4 d4 e4 c4 c4 b3 b3 c4 c4 d4 e4 f4 e4 d4 c4 b4 g3 a3 b3 c4 c4 c4 c4
```

4. Write code for the play_all function. This code should just call play on each instrument in the vector. Remember they are pointers to instruments!

TURN OVER

5. Following the guitar example, create a bass class that inherits from guitar. The play method for the bass should only play every fourth note and skip the others (leave a space), and play the notes one octave lower than the melody (just print out the notes and subtract one from the octaves). This simulating playing on first and fifth beats of every measure only. Here's what the bass looks like alongside the guitar:

Guitar:

c4 c4 d4 e4 c4 e4 d4 g3 c4 c4 d4 e4 c4 c4 b3 b3 c4 c4 d4 e4 f4 e4 d4 c4 b4 g3 a3 b3 c4 c4 c4 c4

Bass:

c3 c3 c3 c3 c3 f3 b3 c3

6. Next, your annoying sibling shows up and wants to join your band. They play violin, but can only play the notes corresponding to the pitches of the four strings on the violin, which are G, D, A, and E. Write a violin class where the play method only plays G's, D's, A's and E's, and skips all other notes. Alongside guitar and bass, here's what your band looks like now:

Guitar:

c4 c4 d4 e4 c4 e4 d4 g3 c4 c4 d4 e4 c4 c4 b3 b3 c4 c4 d4 e4 f4 e4 d4 c4 b4 g3 a3 b3 c4 c4 c4 c4

Bass:

c3 c3 c3 c3 c3 f3 b3 c3

Violin:

d4 e4 e4 d4 g3 d4 e4 d4 e4 e4 d4 g3 a3

7. Fill in the delete_instruments function to delete the instruments since they were created with new.
8. Challenge: add more instruments. For example, add drums that ignore the pitches of the melody, but print B for bass drum and S for snare drum at appropriate times.