**Background:**

Need to have a basic understanding of the following:

* Digital filters
* Frequency response
* Convolution - convolving audio file with the filter to get the filtered output
* MATLAB or equivalent for filter design

This might be a good place to read up on filters:

signal processing & filter design: <http://www.dspguide.com/ch14.htm>

Reference tutorial for voice coil music: <https://touchmysound.wordpress.com/>

**Designing Filters in Matlab:**

1. This tutorial shows steps to create filter using filterDesigner toolbox.

<https://www.mathworks.com/help/signal/gs/designing-the-filter.html>

1. The following tutorial goes into details of filter design using the filterDesigner toolbox including
   * how to export the filter coefficients to Matlab Workspace or
   * how to export filter coefficients into a Matlab file, that can be used in another program.
   * how to auto-generate Matlab code from your filter design.

<https://www.mathworks.com/help/signal/ug/getting-started-with-filter-designer.html#responsive_offcanvas>

9/30: Frequency Response (fft)

Refer to the Noisy signal segment: <https://www.mathworks.com/help/matlab/ref/fft.html>

1. Study and replicate the Matlab code that generates a sinusoidal signal made up of combination of 2 signals at 50Hz and 120Hz.
2. Study the time domain plot.
3. Using the fft function and following the code, generate the frequency response of the signal.

In sptool:

1. Import the signal you generated to sptool and listen to it.
2. Create 2 filters:
   1. one that only removes frequencies below 60Hz
   2. Another one that only removes frequencies above 110Hz
3. Apply the filters to the generated signal one-by-one, and observe the filtered output in time domain. Listen to it too.
4. Now, using the code to generate fft, observe the frequency spectrum of the filtered output.
5. Note your observations.

Assignment#1: Understanding time & frequency domain signals

1. Now, generate a signal made up 3 sinusoidal signals (lets say 200Hz, 1khz, 2kHz).
2. Import to sptool. Observe the time domain signal and listen to it.
3. Plot its frequency response.
4. Create 3 filters that will remove one of the signal components at a time.
5. Observe the filtered output in time and frequency domain.

Assignment#2: Creating an equalizer

**The overall goal:**

Find 5 songs with varied power spectrums and test the ability of individuals to identify the songs through vibrations. This is equivalent to what an equalizer does visually.

In order to do this, we need to represent the frequency (or power) spectrum in terms of vibrations as the song moves forward in time. Thus, we need to determine the frequency spectrum of the song for a small amount of time, then the frequency spectrum for the next time window and then for next...Represent this visually and we have an equalizer.

1. Create 7 different filters:

* Subbass: 0-60Hz,
* Bass: 60-250Hz
* Low midrange: 250-500Hz,
* Midrange: 500-2000Hz
* Upper midrange: 2000-4000Hz,
* Presence: 4000-6000Hz
* Brilliance: 6000-20000Hz

1. Import a song.
2. Take the first 100 samples of the song and convolve this song segment with each filter. Eg. conv(sign1,filter1)
3. The output of each filter gives us the frequency components in that particular frequency range.
4. Within each frequency range (filter output), determine the power spectrum.

i.e. take each of the frequency components, square it and then add all the squares together.

* You should have 7 numbers now: This is the power spectrum of the first few milliseconds of the signal broken down into 7 bands. Save these into 7 different arrays.

1. Move to the next 100 samples and repeat steps 3-5. Keep appending the power spectrum values for each band onto these arrays.
2. After the entire song has been converted into time window specific power spectrums, plot the 7 power spectrum arrays one below the other.