GP 112 ACTIVITY 17

Import the modules

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [ ]:
```
import librosa
import IPython.display as ipd
```

Import the audio

In [ ]:
```
#load .wav sound file
#sr = sample rate = #sam[le per second, 22050 Hz
x, sr = librosa.load('violin-C4.wav')
x = x[0:(3*sr)+1]
print(type(x))
print('x length: {}'.format(len(x)))
print(type(sr))
print('sr = {}'.format(sr))
```

```
<class 'numpy.ndarray'>
x length: 66151
<class 'int'>
sr = 22050
```

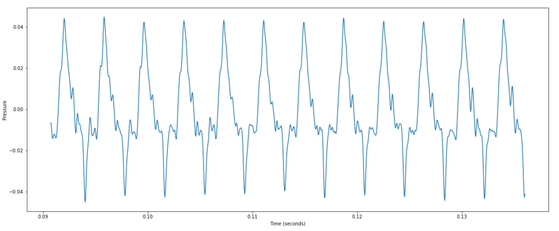
In [ ]:
```
# confirm the audio sample length
print('Sound clip is {} seconds long.'.format((len(x)-1)/sr))
```

```
Sound clip is 3.0 seconds long.
```

Plot the graph in time domain

In [ ]:
```
plt.figure(figsize=(20,8))
# Develop array t to match x
t = np.array(range(0, len(x)))/sr
#plt.plot(t,x)
plt.plot(t[2000:3000],x[2000:3000])
plt.xlabel('Time (seconds)')
plt.ylabel('Pressure')
```

Out[ ]:
```
Text(0, 0.5, 'Pressure')
```



Embad the audio

In [ ]:
```
ipd.Audio(x ,rate = sr)
```

Out[ ]:

Fourier transformation

In [ ]:
```
# find the fourier transform
c = np.fft.fft(x)
print(type(c))
print(len(c))
#print the array of complex numbers
print(c[0:10])
```

```
<class 'numpy.ndarray'>
66151
[-105.52198595-3.10862447e-15j    4.50358546-4.71183184e+00j
   2.83387104-3.87250720e+00j    1.89096091-4.07505836e+00j
   0.64015351-4.07604171e+00j   -0.29418841-3.45535666e+00j
  -0.84512789-2.92568689e+00j   -1.07781596-2.38081172e+00j
  -1.21083966-1.95421703e+00j   -1.26617415-1.64000292e+00j]
```

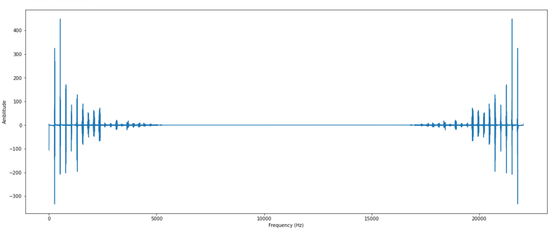
Plot the Frequency vs Amlitude Graph

Graph for whole range

In [ ]:
```
fr = np.array(range(0,66151))/3
# plot the absolute value
plt.figure(figsize=(20,8))
plt.plot(fr,c)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Amblitude')
```

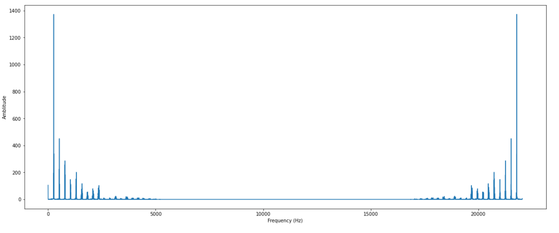
```
c:\Users\Public\anaconda3\lib\site-packages\matplotlib\cbook\__init__.py:1298: ComplexWarning: Casting complex values to real discards the imaginary part
  return np.asarray(x, float)
```

Out[ ]:
```
Text(0, 0.5, 'Amblitude')
```



In [ ]:
```
fr = np.array(range(0,66151))/3
# plot the absolute value
plt.figure(figsize=(20,8))
plt.plot(fr,np.abs(c))
plt.xlabel('Frequency (Hz)')
plt.ylabel('Amblitude')
```

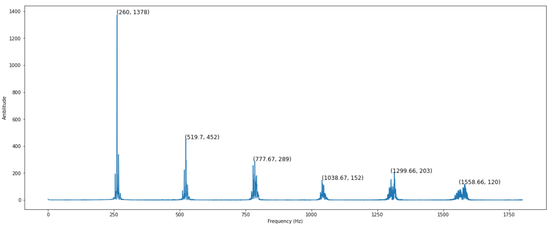
Out[ ]:
```
Text(0, 0.5, 'Amblitude')
```



Graph for specific range

In [ ]:
```
fig, ax = plt.subplots(figsize=(20,8))
condition = (fr >0) & (fr < 1800)
plt.plot(fr[condition],np.abs(c[condition]))
plt.xlabel('Frequency (Hz)')
plt.ylabel('Amblitude')
ax.text(260, 1378, '(260, 1378)', size=12)
ax.text(519.7, 452, '(519.7, 452)', size=12)
ax.text(777.67, 289, '(777.67, 289)', size=12)
ax.text(1038.67, 152, '(1038.67, 152)', size=12)
ax.text(1299.66, 203, '(1299.66, 203)', size=12)
ax.text(1558.66, 120, '(1558.66, 120)', size=12)
```

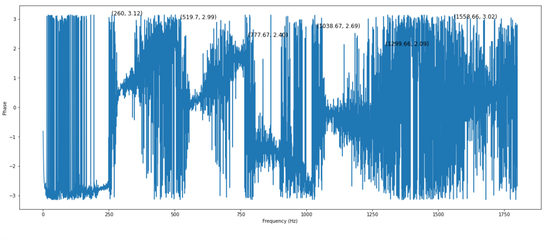
Out[ ]:
```
Text(1558.66, 120, '(1558.66, 120)')
```



Plot Frequency vs Phase graph

In [ ]:
```
fig, ax = plt.subplots(figsize=(20,8))
condition = (fr >0) & (fr < 1800)
plt.plot(fr[condition],np.angle(c[condition]))
plt.xlabel('Frequency (Hz)')
plt.ylabel('Phase')
ax.text(260, 3.12, '(260, 3.12)', size=12)
ax.text(519.7, 2.99, '(519.7, 2.99)', size=12)
ax.text(777.67, 2.40, '(777.67, 2.40)', size=12)
ax.text(1038.67, 2.69, '(1038.67, 2.69)', size=12)
ax.text(1299.66, 2.09, '(1299.66, 2.09)', size=12)
ax.text(1558.66, 3.02, '(1558.66, 3.02)', size=12)
```

Out[ ]:
```
Text(1558.66, 3.02, '(1558.66, 3.02)')
```



Zoomed frequency vs phase graph

In [ ]:
```
fig, ax = plt.subplots(figsize=(20,8))
condition = (fr >1555) & (fr < 1565)
plt.plot(fr[condition],np.angle(c[condition]))
plt.xlabel('Frequency (Hz)')
plt.ylabel('Phase')
ax.text(1558.66, 3.02, '(1558.66, 3.02)', size=12)
```

Out[ ]:
```
Text(1558.66, 3.02, '(1558.66, 3.02)')
```