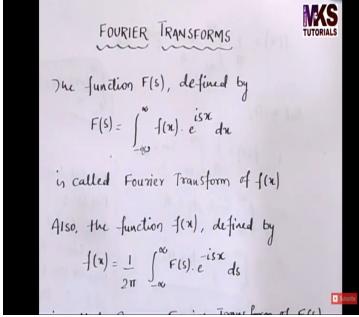
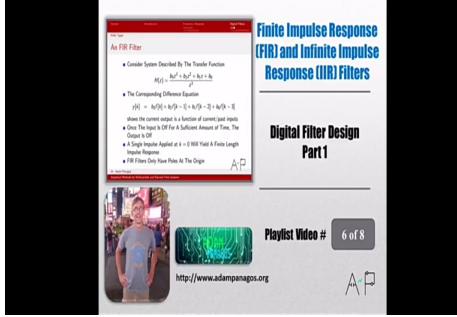
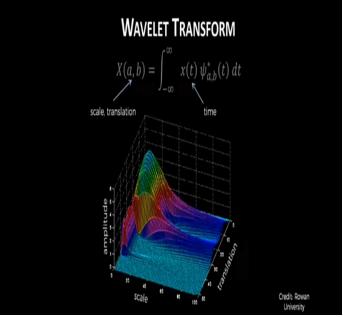


DAY 8 Report

Date:	27/05/20	Name:	Ankitha c c
Course:	Digital signal processing	USN:	4al16ec004
Topic:	1. Fourier transformation 2. FFT, FFT IN MATLAB 3. FIR AND IIR Filter 4. Welch's methods and winnowing	Semester & Section:	8th & a
Github Repository:	ankitha-course		

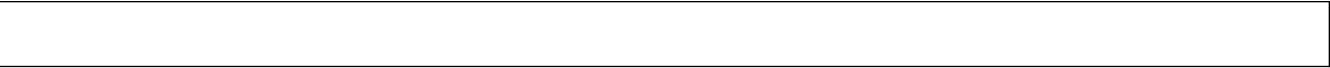
FORENOON SESSION DETAILS	
Image of session	  
Report – Report can be typed or hand written for up to two pages.	<p>Gaussian functions</p> <pre> Fs = 100; % Sampling frequency t = -0.5:1/Fs:0.5; % Time vector L = length(t); % Signal length X = 1/(4*sqrt(2*pi*0.01))*exp(-t.^2/(2*0.01)); plot(t,X) </pre>

```
title('Gaussian Pulse in Time Domain')  
xlabel('Time (t)')  
ylabel('X(t)')
```

Cosine functions

```
Fs = 1000; % Sampling frequency  
T = 1/Fs; % Sampling period  
L = 1000; % Length of signal  
t = (0:L-1)*T;  
  
x1 = cos(2*pi*50*t); % First row wave  
x2 = cos(2*pi*150*t); % Second row wave  
x3 = cos(2*pi*300*t); % Third row wave  
  
X = [x1; x2; x3];
```

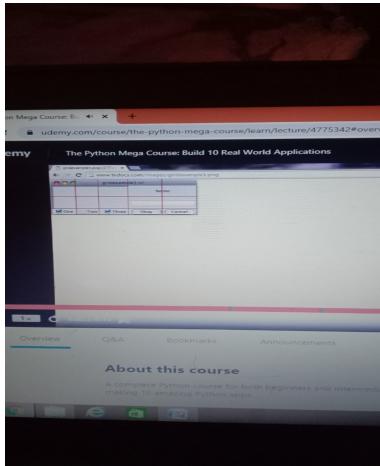
```
for i = 1:3  
    subplot(3,1,i)  
    plot(t(1:100),X(i,1:100))  
    title(['Row ',num2str(i),' in the Time Domain'])  
end
```



Date: 27/05/20
Name: Ankitha C C
Course: Python USN: 4AL16EC004
Topic: 1. Graphical user interface with tkinter Semester & 8th & a
2. Interacting with database Section:

AFTERNOON SESSION DETAILS

Image of session



```
L75: Connecting and Inserting Data to SQLite via Python

# coding: utf-8
# import the sqlite3 module
import sqlite3

# create a database connection
conn = sqlite3.connect('myDB.db')

# create a cursor object
cursor = conn.cursor()

# execute a query
cursor.execute("SELECT * FROM Books")

# fetch all the rows
rows = cursor.fetchall()

# print the rows
for row in rows:
    print(row)

# close the connection
conn.close()
```

```
SQLite3 Database - Books Table

SELECT * FROM Books
1| "The Alchemist", "Paulo Coelho", 10.99, 100, 1000
2| "The Hobbit", "J.R.R. Tolkien", 15.99, 200, 2000
3| "The Lord of the Rings", "J.R.R. Tolkien", 15.99, 200, 2000
4| "The Catcher in the Rye", "J.D. Salinger", 12.99, 100, 1000
5| "To Kill a Mockingbird", "Harper Lee", 12.99, 100, 1000
6| "The Great Gatsby", "F. Scott Fitzgerald", 12.99, 100, 1000
7| "1984", "George Orwell", 12.99, 100, 1000
8| "The Handmaid's Tale", "Margaret Atwood", 12.99, 100, 1000
9| "The Road", " Cormac McCarthy", 12.99, 100, 1000
10| "The Pillars of the Earth", "Ken Follett", 12.99, 100, 1000

# close the connection
conn.close()
```



Report - Report can be typed or hand written for up to two pages.

Create a Multi-widget GUI (Practice)

Create a Python program that expects a kilogram input value and converts that value to grams, pounds, and ounces when the user pushes the Convert button.

The program will look similar to the one in the following picture:

Tip:

1 kg = 1000 grams

1 kg = 2.20462 pounds

1 kg = 35.274 ounces

Solution

```
from tkinter import *
# Create an empty Tkinter window
window=Tk()
def from_kg():
    # Get user value from input box and multiply by 1000 to get kilograms
    gram=float(e2_value.get())*1000
    # Get user value from input box and multiply by 2.20462 to get pounds
    pound=float(e2_value.get())*2.20462
    # Get user value from input box and multiply by 35.274 to get ounces
    ounce=float(e2_value.get())*35.274
    # Empty the Text boxes if they had text from the previous use and fill them again
    t1.delete("1.0", END) # Deletes the content of the Text box from start to END
    t1.insert(END,gram) # Fill in the text box with the value of gram variable
    t2.delete("1.0", END)
    t2.insert(END,pound)
    t3.delete("1.0", END)
    t3.insert(END,ounce)
# Create a Label widget with "Kg" as label
```

```

e1=Label(window,text="Kg")
e1.grid(row=0,column=0) # The Label is placed in position 0, 0 in the window
e2_value=StringVar() # Create a special StringVar object
e2=Entry(window,textvariable=e2_value) # Create an Entry box for users to enter the value
e2.grid(row=0,column=1)
# Create a button widget
# The from_kg() function is called when the button is pushed
b1=Button(window,text="Convert",command=from_kg)
b1.grid(row=0,column=2)

# Create three empty text boxes, t1, t2, and t3
t1=Text(window,height=1,width=20)
t1.grid(row=1,column=0)

t2=Text(window,height=1,width=20)
t2.grid(row=1,column=1)

t3=Text(window,height=1,width=20)
t3.grid(row=1,column=2)

# This makes sure to keep the main window open
window.mainloop()

```

Querying data from a MySQL database

In previous videos I explained how to interact with PostGreSQL databases. If you prefer to work with MySQL instead of PostGreSQL, see the code further down.

I set up a remote MySQL database on a server with the IP address 108.167.140.122, so you don't have to install and set up a MySQL database yourself. To connect and query data from that remote database, you need a username, password, and the name of the database. These are written inside the Python s

cript below.

You also need a Python library that interacts with MySQL databases. Many libraries are compatible, but I prefer mysql.connector. To install mysql.connector: simply execute pip install mysql-connector or pip3 install mysql-connector depending on whether you use pip or pip3. Once you install the library, try this working example:

```
import mysql.connector
word = input("Enter a word in English and press Enter:")
con = mysql.connector.connect(
    user="ardit700_student",
    password = "ardit700_student",
    host="108.167.140.122",
    database = "ardit700_pm1database"
)
cursor = con.cursor()
query = cursor.execute("SELECT * FROM Dictionary WHERE Expression = '%s'" % word)
results = cursor.fetchall()
if results:
    for result in results:
        print(result[1])
else:
    print("We couldn't find any results about that.")
```