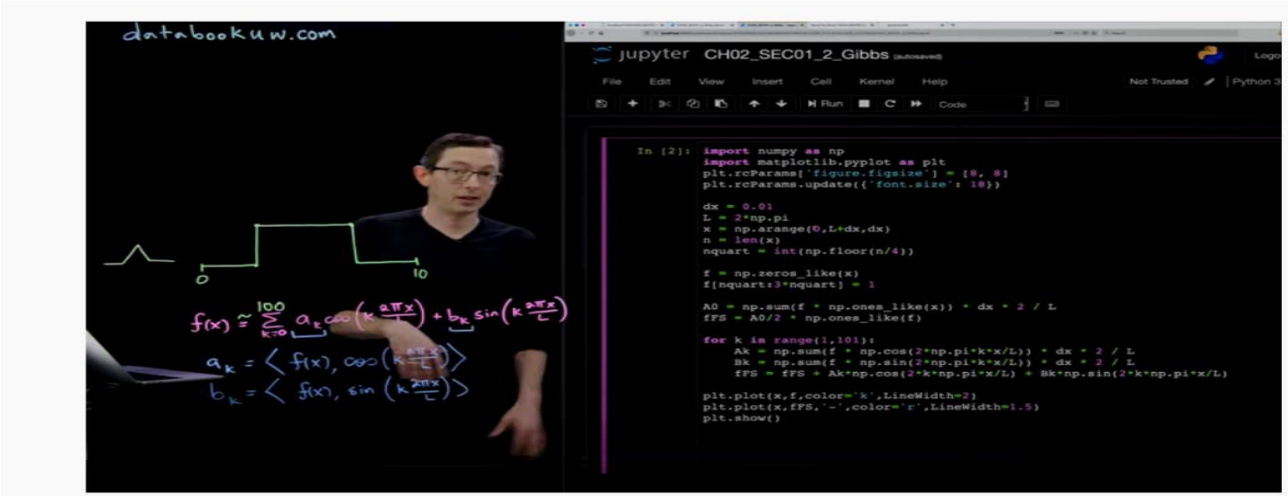
-DAILY ASSESSMENT FORMAT

Date:	**26-05-2020**	**Name:**	**Kishan shetty**
Course:	**Digital signal processing**	**USN:**	**4AL17EC093**
Topic:	**Fourier Series & Gibbs Phenomena using Python Fourier Transform and derivatives convolution, laplace transform and z- transform**	**Semester & Section:**	**6th & 'a'**
GitHub Repository:	**Kishanshetty-041**		

FORENOON SESSION DETAILS
Image of session **1.** 
Report – Report can be typed or hand written for up to two pages. Derivatives of functions The Fourier transform of the derivative of a function is given by:

$$\begin{aligned}\mathcal{F}\left(\frac{d}{dx}f(x)\right) &= \int_{-\infty}^{\infty} \overbrace{f'(x)}^{dv}\overbrace{e^{-i\omega x}}^{u}\,dx \\ &= \Big[\underbrace{f(x)e^{-i\omega x}}_{uv}\Big]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \underbrace{f(x)}_{v}\Big[\underbrace{-i\omega e^{-i\omega x}}_{du}\Big]\,dx \\ &= i\omega\int_{-\infty}^{\infty} f(x)e^{-i\omega x}\,dx \\ &= i\omega\mathcal{F}(f(x)).\end{aligned}$$

• This is an extremely important property of the Fourier transform, as it will allow us to turn PDEs into ODEs, closely related to the separation of variables:

$$\underbrace{u_{tt} = cu_{xx}}_{\text{(PDE)}} \quad \overset{\mathcal{F}}{\Longrightarrow} \quad \underbrace{\hat{u}_{tt} = -c\omega^2\hat{u}.}_{\text{(ODE)}}$$

Linearity of Fourier transforms The Fourier transform is a linear operator, so that:

$$\mathcal{F}(\alpha f(x) + \beta g(x)) = \alpha\mathcal{F}(f) + \beta\mathcal{F}(g).$$

$$\mathcal{F}^{-1}(\alpha\hat{f}(\omega) + \beta\hat{g}(\omega)) = \alpha\mathcal{F}^{-1}(\hat{f}) + \beta\mathcal{F}^{-1}(\hat{g}).$$

Parseval's theorem

$$\int_{-\infty}^{\infty} |\hat{f}(\omega)|^2\,d\omega = 2\pi\int_{-\infty}^{\infty} |f(x)|^2\,dx.$$

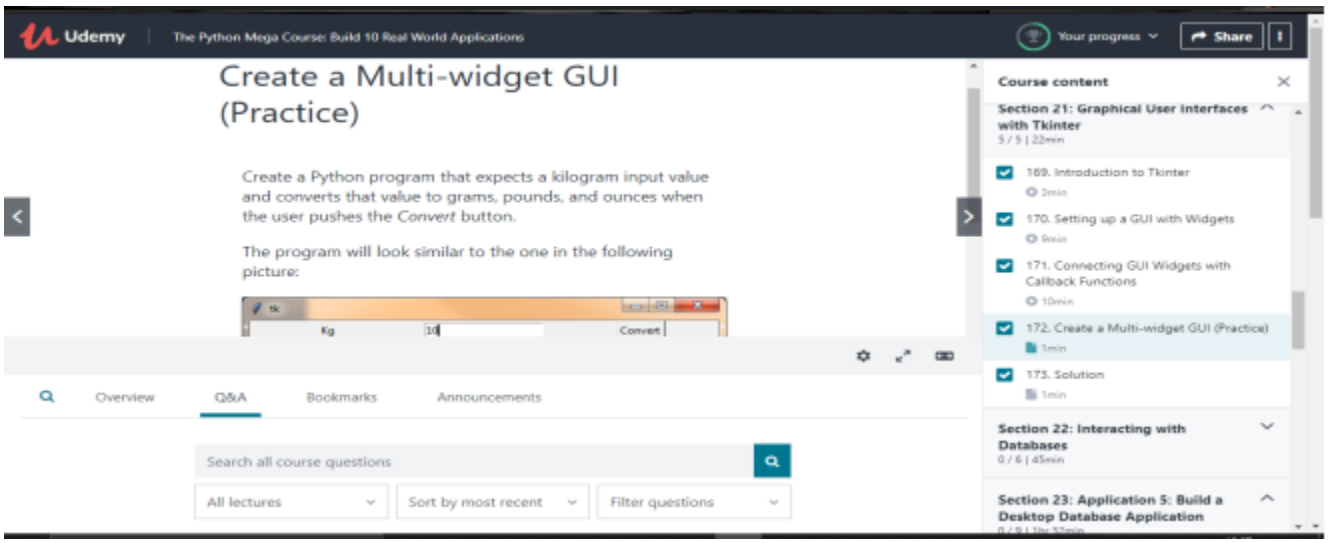
Convolution The convolution of two functions is particularly well-behaved in the Fourier domain, being the product of the two Fourier transformed functions. Define the convolution of two functions f(x) and g(x) as f * g:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(x-\xi)g(\xi)\,d\xi.$$

If we let ^ f = F(f) and ^g = F(g), then:

$$
\begin{aligned}
\mathcal{F}^{-1}\left(\hat{f}\hat{g}\right)(x) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\omega)\hat{g}(\omega)e^{i\omega x}\,d\omega \\
&= \int_{-\infty}^{\infty}\hat{f}(\omega)e^{i\omega x}\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}g(y)e^{-i\omega y}\,dy\right)d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}g(y)\hat{f}(\omega)e^{i\omega(x-y)}\,d\omega\,dy \\
&= \int_{-\infty}^{\infty}g(y)\underbrace{\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\omega)e^{i\omega(x-y)}\,d\omega\right)}_{f(x-y)}dy \\
&= \int_{-\infty}^{\infty}g(y)f(x-y)\,dy = g*f = f*g.
\end{aligned}
$$

Date:	26/05/2020	Name:	Kishan shetty
course	python	usn	4al17ec041
topic	: 1] Graphical User Interfaces with Tkinter Semester & Section	Semester & Section:	: 6 th 'A'
Github Repository	Kishanshetty-041		



Report

Graphical User Interfaces with Tkinter

- The graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.

- Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

- The two main elements of GUI are Window and Widgets.

- A window is an area on the screen that displays information, with its contents being displayed independently from the rest of the screen.

- Interface elements known as graphical control elements, controls or widgets are software components that a computer user interacts with through direct manipulation to read or edit information about an application. Each widget facilitates a specific user-computer interaction.

- The Tk () function is used to create a GUI window.

- The mainloop () function is used to keep GUI window open until user closes window mandatorily.

- To create widgets:

    - The Button () function is used to implement various kinds of buttons. Buttons can contain text or images, and you can associate a Python function or method with each button. When the button is pressed, Tkinter automatically calls that function or method.
    - The Entry () function is used to accept single-line text strings from a user.
    - The text () function is used to display text documents, containing either plain text or formatted text (using different fonts, embedded images, and other embellishments). The text widget can also be used as a text editor.