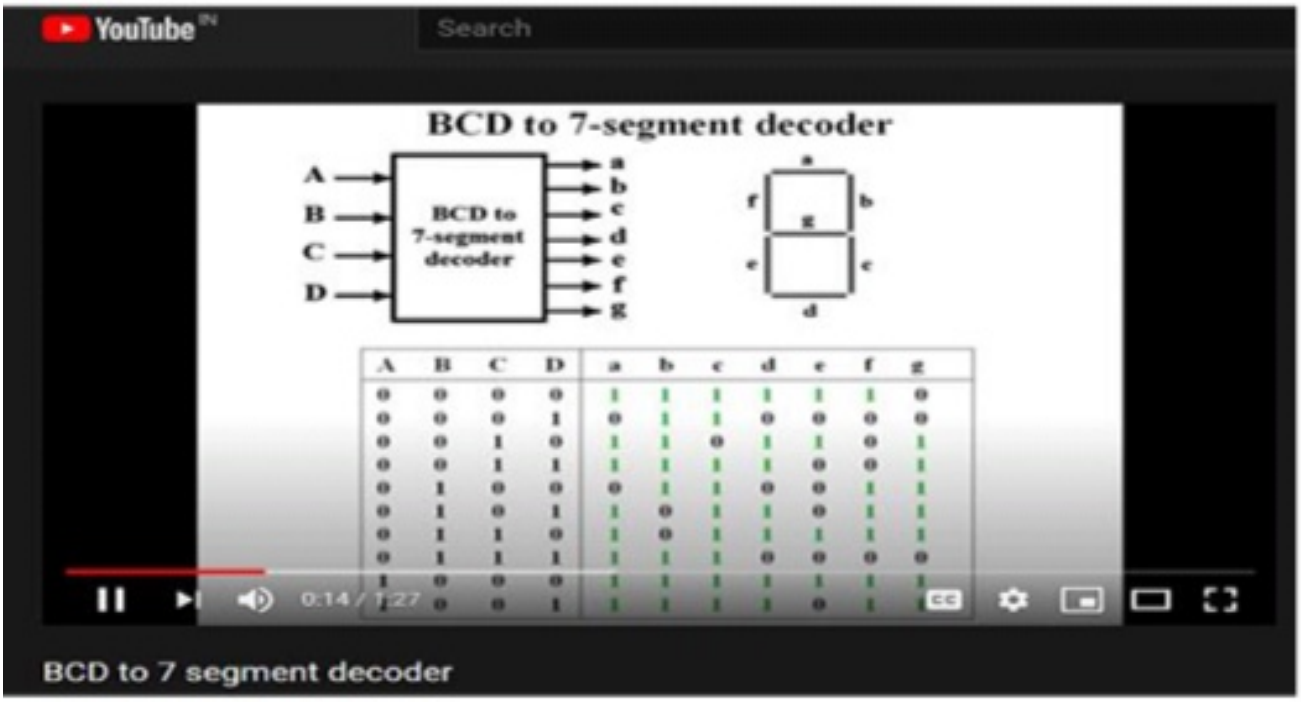
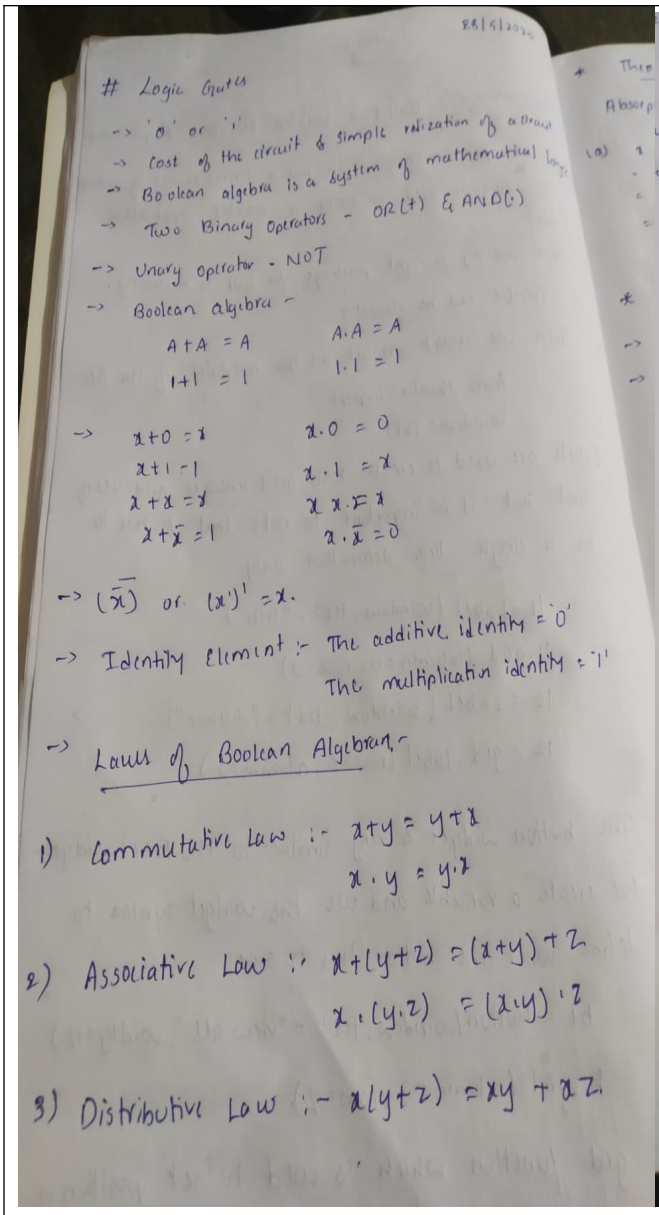
DAILY ASSESSMENT FORMAT

Date:	28-05-2020	Name:	M V Ramya
Course:	DSP	USN:	4AL17EC045
Topic:	Fourier transform	Semester & Section:	6th sem, A sec
Github Repository:	M V Ramya-045		

FORENOON SESSION DETAILS





Logic Gates

-> '0' or '1'
-> Cost of the circuit & simple realization of a circuit
-> Boolean algebra is a system of mathematical logic
-> Two Binary operators - OR(+) & AND(.)
-> Unary operator - NOT
-> Boolean algebra -

$A+A = A$ $\quad A.A = A$
$1+1 = 1$ $\quad 1.1 = 1$

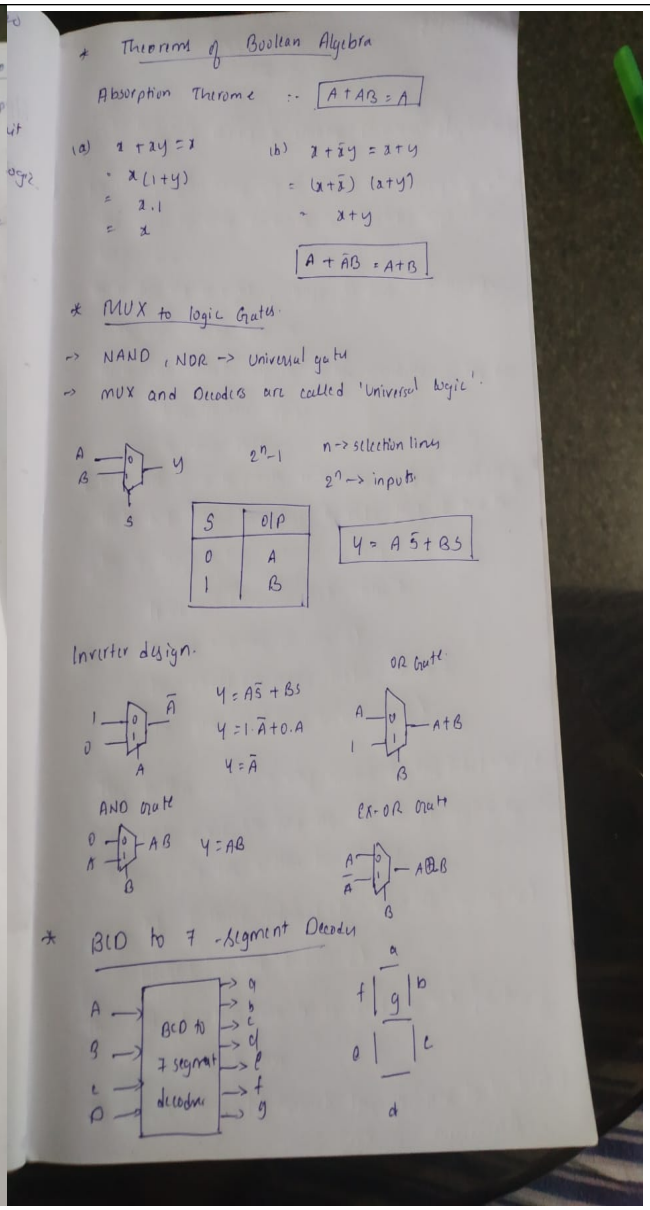
-> $x+0 = x$ $\quad x.0 = 0$
$x+1 = 1$ $\quad x.1 = x$
$x+x = x$ $\quad x.x = x$
$x+\bar{x} = 1$ $\quad x.\bar{x} = 0$

-> $(\bar{x})$ or $(x')' = x$.
-> Identity Element :- The additive identity = '0'
The multiplication identity = '1'
-> Laws of Boolean Algebra :-

1) Commutative Law :- $x+y = y+x$
$x.y = y.x$

2) Associative Law :- $x+(y+z) = (x+y)+z$
$x.(y.z) = (x.y).z$

3) Distributive Law :- $x(y+z) = xy + xz$



* Theorems of Boolean Algebra

Absorption Theorem :- $\boxed{A+AB = A}$

(a) $x+xy = x$
$= x(1+y)$
$= x.1$
$= x$

(b) $x+\bar{x}y = x+y$
$= (x+\bar{x})(x+y)$
$= x+y$

$\boxed{A+\bar{A}B = A+B}$

* MUX to logic Gates

-> NAND, NOR -> Universal gate
-> MUX and Decoders are called 'Universal logic'.

2^n-1
n -> selection lines
2^n -> inputs

S	O/P
0	A
1	B

$\boxed{Y = A\bar{S}+BS}$

Inverter design.

$Y = A\bar{S}+BS$
$Y = 1.\bar{A}+0.A$
$Y = \bar{A}$

OR Gate

AND Gate

$Y = AB$

Ex-OR Gate

* BCD to 7-Segment Decoder

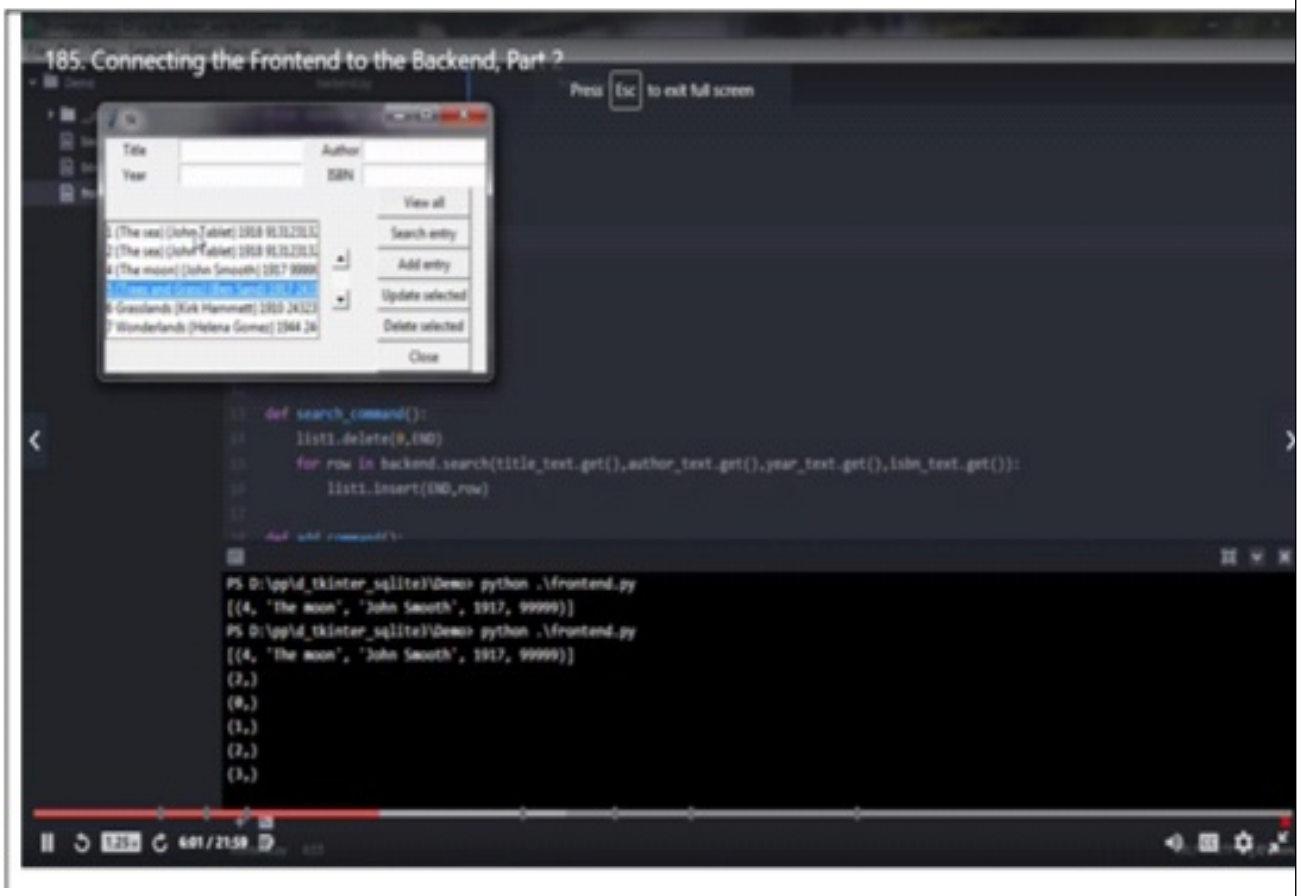
Date:28may2020 Name: MV Ramya

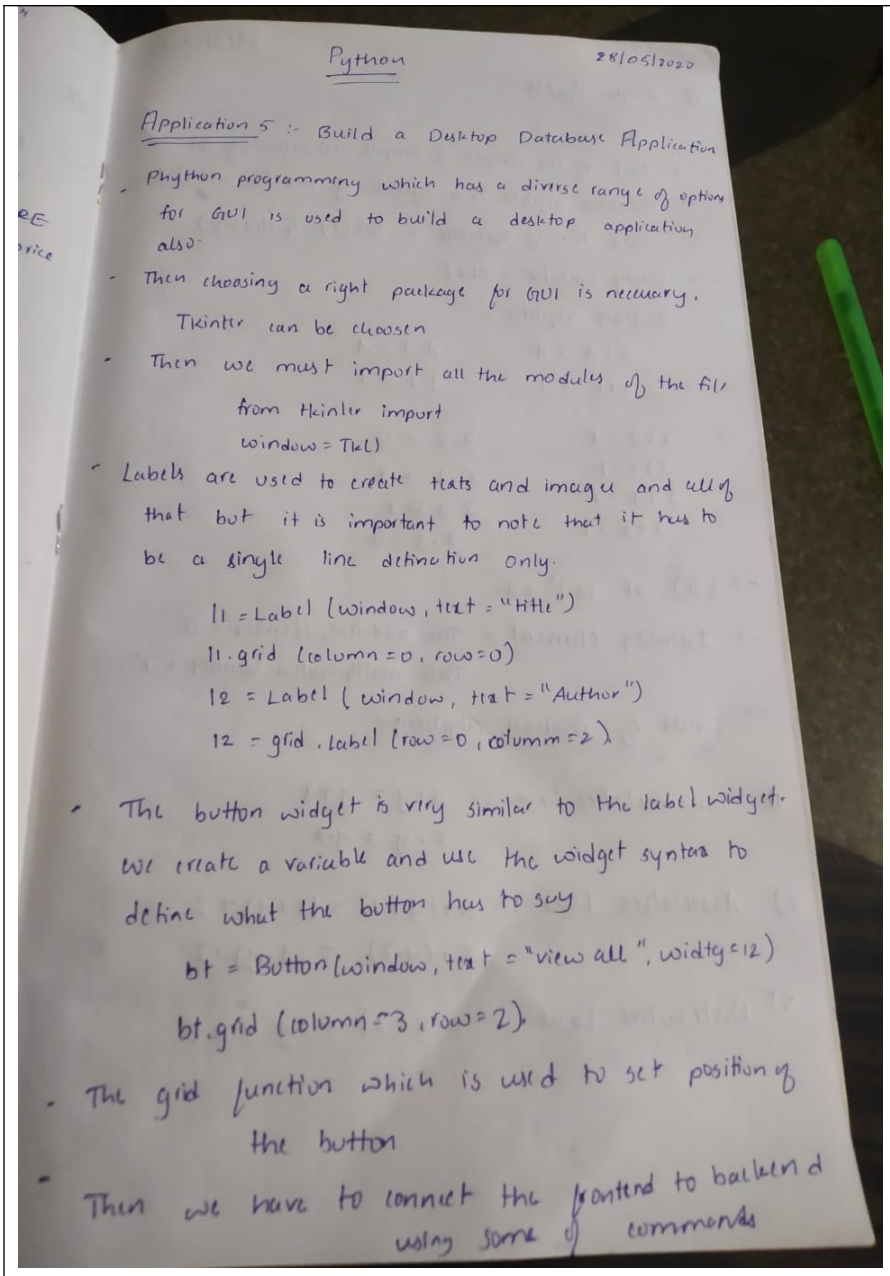
Course: python USN:4AL17EC045

Topic: python Semester & 6th sem Asec

Section:

AFTERNOON SESSION DETAILS
Image of session





Python 28/05/2020

Application 5 :- Build a Desktop Database Application

- Phython programming which has a diverse range of options for GUI is used to build a desktop application also
- Then choosing a right package for GUI is neccesary. Tkinter can be choosen
- Then we must import all the modules of the file

```
from tkinter import
window = Tkl()
```

- Labels are used to create texts and images and all of that but it is important to note that it has to be a single line definition only.

```
l1 = Label (window, text = "Title")
l1.grid (column = 0, row = 0)
l2 = Label (window, text = "Author")
l2 = grid.Label (row = 0, columm = 2)
```

- The button widget is very similar to the label widget. We create a variable and use the widget syntax to define what the button has to say

```
bt = Button (window, text = "view all", widtg = 12)
bt.grid (column = 3, row = 2)
```

- The grid function which is used to set position of the button
- Then we have to connect the frontend to backend using some commands