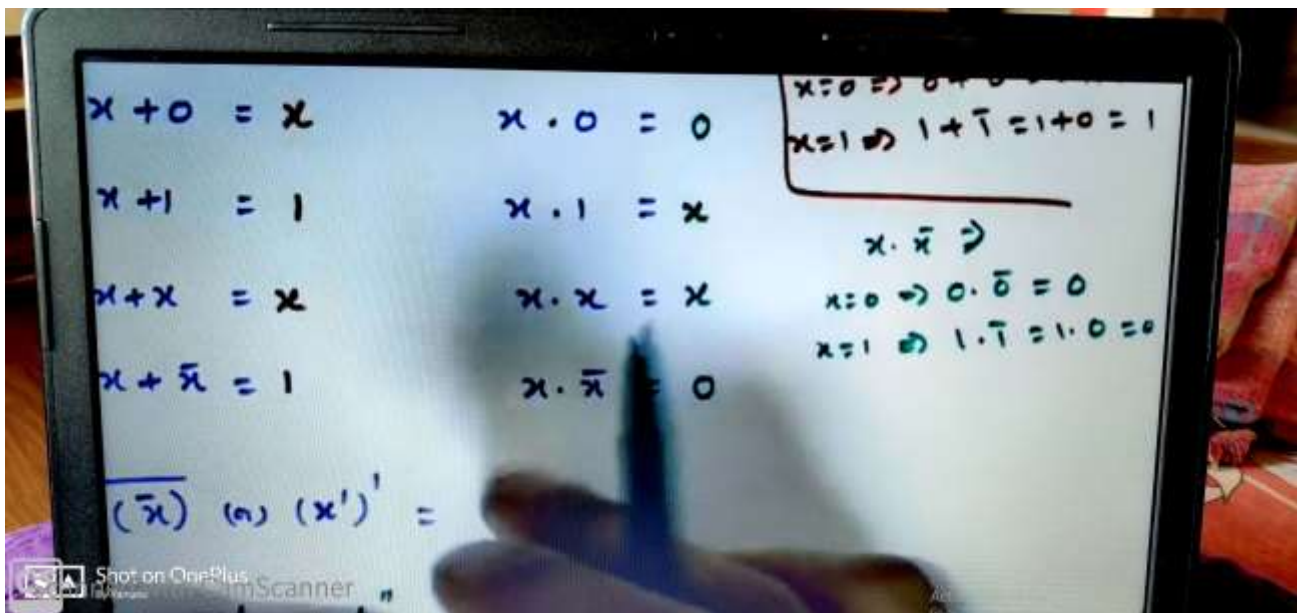
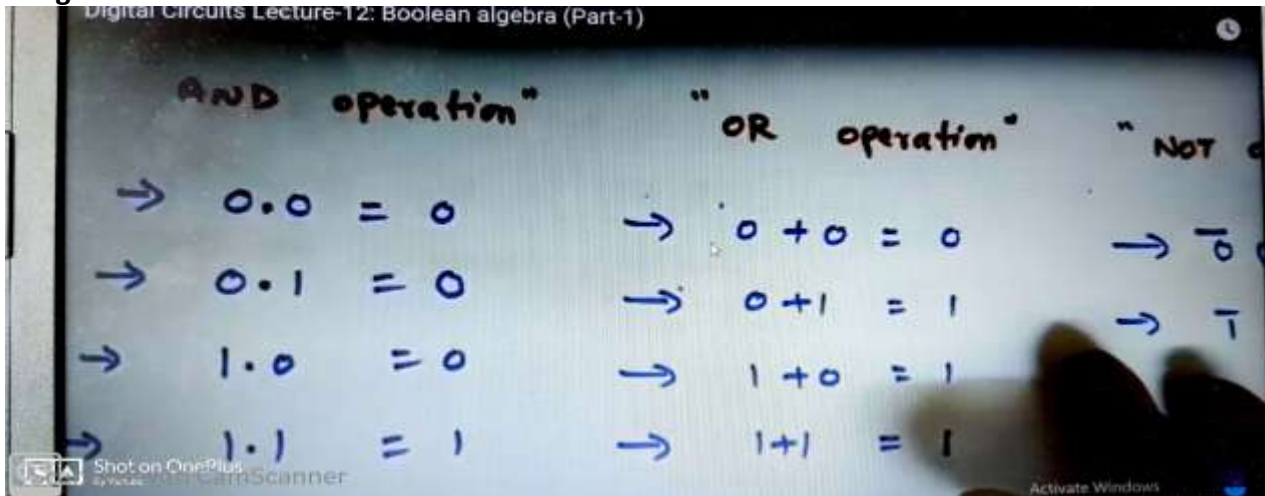


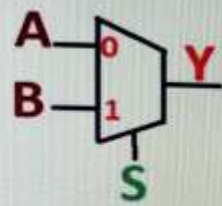
## DAILY ASSESSMENT FORMAT

Date:	28-05-2020	Name:	POOJA K S
Course:	Logic Design	USN:	4AL17EC070
Topic:	Boolean equations for digital circuit. Combinational circuits: conversion of MUX and decoders to logic gates. Design 7 segment decoder with common anode display.	Semester & Section:	6 <sup>th</sup> B sec
Github Repository:	pooja-shivanna		

### FORENOON SESSION DETAILS

Image of session





SELECTION (S)	OUTPUT(Y)
0	A
1	B

$$2^n - 1$$

$$2^n = \text{inputs}$$

$n$  = selection lines

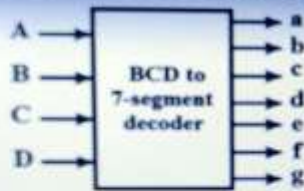
$$Y = A\bar{S} + BS$$



Shot on OnePlus  
Scanned with CamScanner

BCD to 7 segment decoder

BCD to 7-segment decoder



A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1



Shot on OnePlus  
Scanned with CamScanner

## Logic Design

### Boolean equations for digital circuits

- Cost of the circuit
- Simple realization of a circuit
- Boolean algebra is a system of mathematical
- It is defined with set of elements, a set of operators, and a number of axioms or postulates
- set of elements (0, 1)
- binary operators - or, and
- unary " - NOT

### Axioms & Laws of Boolean Algebra

Difference between boolean algebra, ordinary algebra and binary number system

#### Boolean Algebra :-

$$A + A = A \quad A \cdot A = A$$

$$1 + 1 = 1 \quad 1 \cdot 1 = 1$$

#### Ordinary Algebra

$$A + A = 2A \quad A \cdot A = A^2$$

$$1 + 1 = 2 \quad 1 \cdot 1 = 1$$

#### Binary number system

$$1 + 1 = (10) \quad 1 \cdot 1 = 1$$

### Axioms and Postulates :-

$$\rightarrow (x')' = x$$

#### Identity Element :-

or operation

additive identity = 0

multiplicative = 1

AND operation

#### Laws of Boolean Algebra :-

$$1) \text{ Commutative Law :- } x + y = y + x \quad x \cdot y = y \cdot x$$

$$A + B = B + A \quad A \cdot B = B \cdot A$$

#### MUX to logic gates :-

NAND, NOR → Universal gates  
MUX and Decoders are called "Universal logic"

#### Switcher Design

$$Y = 1 \cdot \bar{A} + 0 \cdot A$$

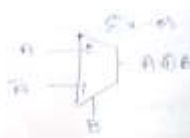
$$Y = \bar{A}$$

#### AND gate

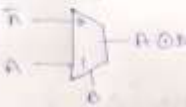
$$Y = 0 \cdot \bar{B} + A \cdot B$$

$$Y = AB$$

#### OR gate



#### Ex - NOR



#### Bcd to 7 segment decoder

dec	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	0	0	1	1	1
1	0	1	1	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	1	0	0
3	1	1	0	1	0	0	0	0	0	0
4	1	1	0	0	1	0	0	0	0	0
5	1	1	0	0	0	1	0	0	0	0
6	1	1	0	0	0	0	1	0	0	0
7	1	1	1	1	1	1	0	0	0	0
8	1	1	1	1	0	1	1	1	1	1
9	1	1	1	1	0	1	1	1	0	1

$$a = \bar{C} + A + B\bar{D} + BD$$

$$b = \bar{B} + CD + \bar{E}\bar{D}$$

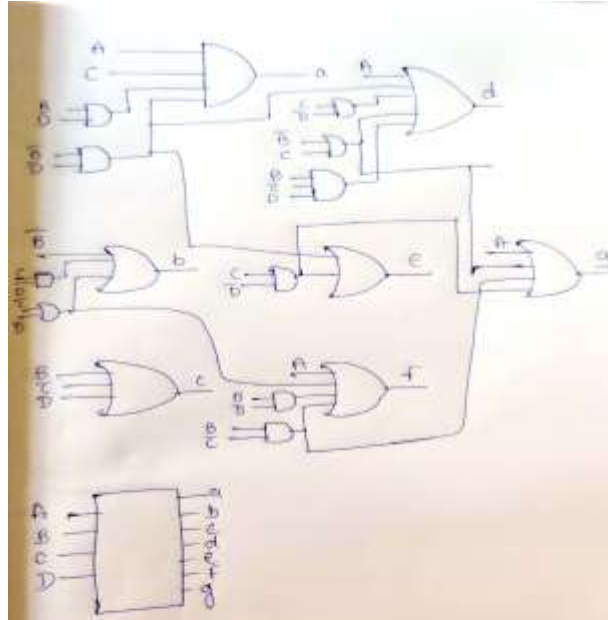
$$c = B + \bar{E} + D$$

$$d = A + B\bar{D} + \bar{C}\bar{D} + \bar{B}C + B\bar{C}\bar{D}$$

$$e = \bar{B}\bar{D} + \bar{C}\bar{D}$$

$$f = A + \bar{E}\bar{D} + B\bar{D} + \bar{B}\bar{C}$$

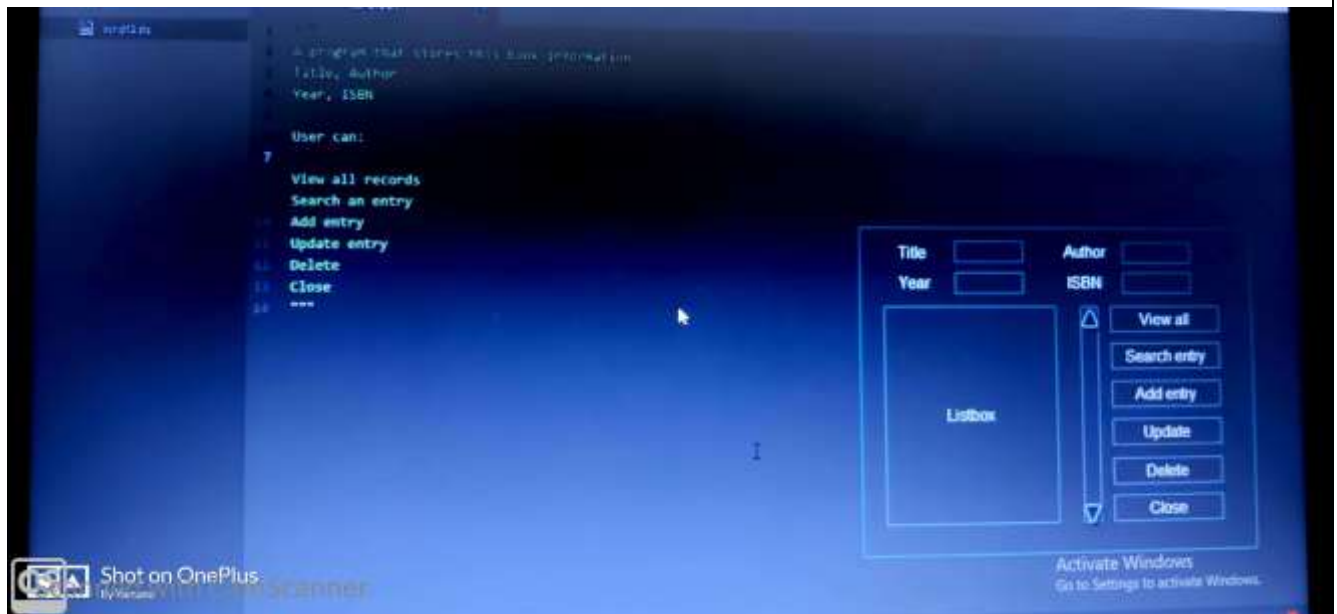
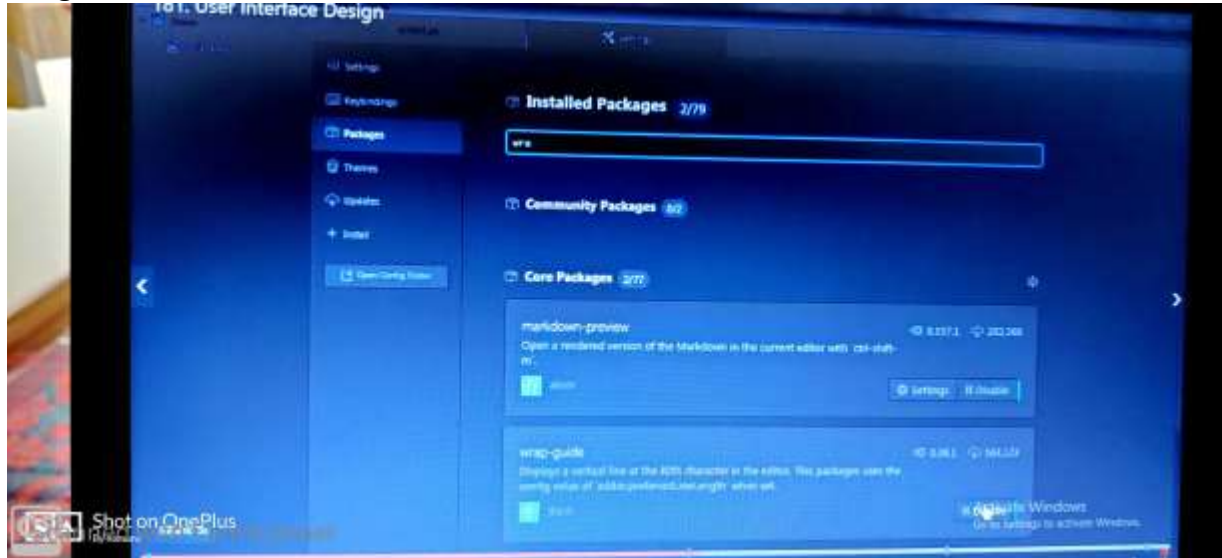
$$g = A + B\bar{C} + \bar{B}C + \bar{C}\bar{D}$$



Date:	28-05-2020	Name:	POOJA K S
Course:	Python Programming	USN:	4AL17EC070
Topic:	Application 5: Build a desktop database application.	Semester & Section:	6 <sup>th</sup> sem B sec

#### AFTERNOON SESSION DETAILS

##### Image of session





## 184. Connecting the Frontend to the Backend, Part 1

```

1 from tkinter import *
2 from backend import *
3
4 def view_command():
5     list.delete(0, list.size()-1)
6     for row in backend.view():
7         list.insert(END, row)
8
9 window=Tk()
10
11 list=Listbox(window, text="Title")
12 list.grid(row=0, column=0)
13
14 list=Listbox(window, text="Author")
15 list.grid(row=0, column=1)
16
17 list=Listbox(window, text="Year")
18 list.grid(row=0, column=2)
19
20 view_button=Button(window, text="View", command=view_command)
21 view_button.grid(row=1, column=0)
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

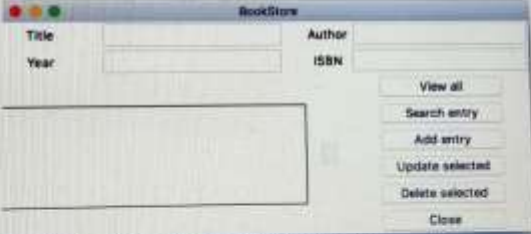
```

Traceback (most recent call last):  
 File "C:\Users\Warius.Grete-PC\AppData\Local\Programs\Python\Python35-32\lib\tkinter\\_init\_.py", line 1549, in \_call\_\_  
 return self.func(\*args)  
 File "frontend.py", line 5, in view\_command  
 for row in backend.view():  
NameError: name 'backend' is not defined  
PS D:\app\id\_tkinter\_sqlite3\Demo> python frontend.py  
[('The sea', 'John Tablet', 1918, 913523132), (2, 'The sea', 'John Tablet', 1918, 913523132), (4, 'The Active Life', 'John Tablet', 1918, 913523132), (6, 'The Active Life', 'John Tablet', 1918, 913523132)]

## Fixing the Bug (Practice)

### Exercise

If you haven't already noticed, the program has a bug. When the listbox is empty and the user clicks the listbox, an `IndexError` is generated in the terminal:



```

AddItD-60P:ap05-tkinter-sqlite wia& python3 frontend.py
Exception in tkinter callback
Traceback (most recent call last):
  File "C:\Users\Warius.Grete-PC\AppData\Local\Programs\Python\Python35-32\lib\tkinter\_init_.py", line 1549, in _call__
    return self.func(*args)
  File "frontend.py", line 4, in get_selected_row
    index=list1.curselection()[0]
IndexError: tuple index out of range

```

## Application 5: Build a Desktop Database

### Application:

A program that stores this book information:

Title: Author

Year: ISBN

User can:

view all records

search an entry

add entry

update entry

Delete

Close

import sqlite3

def connect():

conn = sqlite3.connect("books.db")

cur = conn.cursor()

cur.execute("Create Table if Not exists book")

(id Integer Primary Key, title text, author text, year integer, isbn integer)")

conn.commit()

conn.close()

def insert(title, author, year, isbn):

conn = sqlite3.connect("books.db")

cur = conn.cursor()

cur.execute("Insert into book values (Null, ?, ?, ?, ?)")

(title, author, year, isbn)

conn.commit()

conn.close()

def view():

conn = sqlite3.connect("books.db")

cur = conn.cursor()

cur.execute("select \* from book")

rows = cur.fetchall()

conn.close()

return rows

Solution:-

def get\_selected\_row(event):

try:

global selected\_tuple

index = list1.curselection()[0]

selected\_tuple = list1.get(index)

e1 = delete(0, END)

e1.insert(END, selected\_tuple[1])

e2 = delete(0, END)

e2.insert(END, selected\_tuple[2])

e3 = delete(0, END)

e3.insert(END, selected\_tuple[3])

e4 = delete(0, END)

e4.insert(END, selected\_tuple[4])

except IndexError:

pass