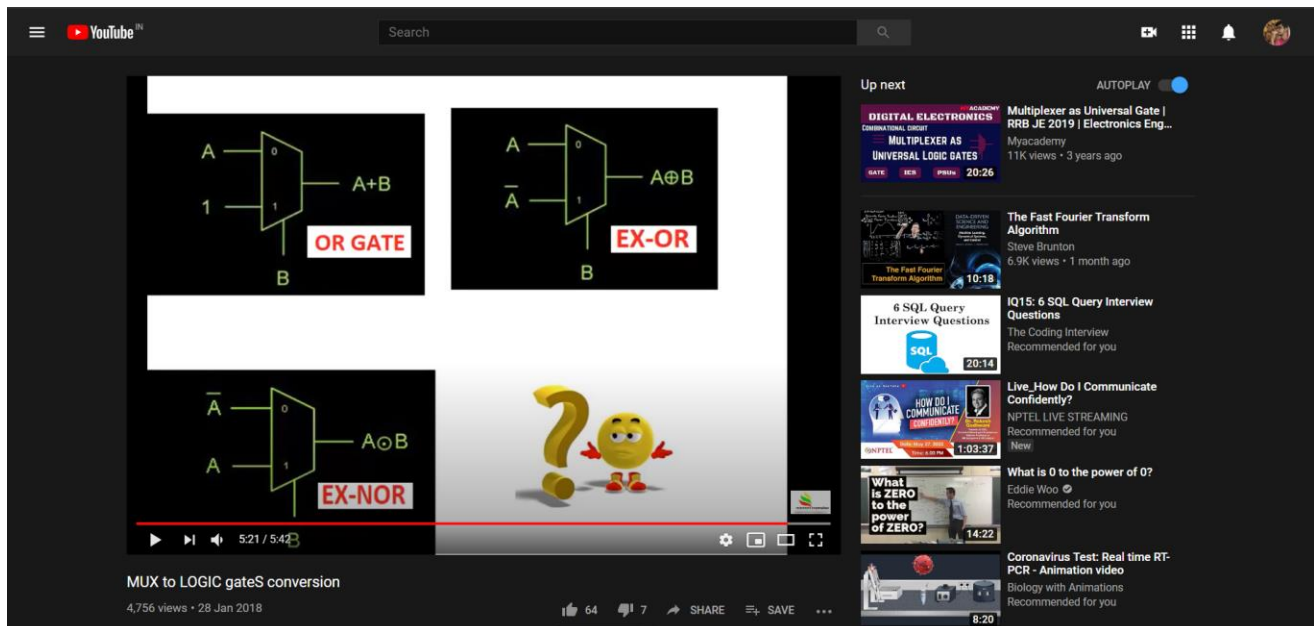


## DAILY ASSESSMENT REPORT

Date:	28 May 2020	Name:	Gagan M K
Course:	LOGIC DESIGN	USN:	4AL17EC032
Topic:	<ul style="list-style-type: none"> <li>Boolean equations for digital circuits.</li> <li>Combinational circuits: Conversion of MUX and Decoders to logic gates.</li> <li>design of 7 segment decoder with common anode display</li> </ul>	Semester & Section:	6 <sup>th</sup> sem & 'A' sec
Github Repository:	Alvas-education-foundation/Gagan-Git		

### FORENOON SESSION DETAILS

Image of session



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

### Boolean equations for digital circuits :

- Digital Circuits - Boolean algebra. Boolean algebra is an algebra, which deals with binary numbers & binary variables. Hence, it is also called as Binary Algebra or logical Algebra.
- The variables used in this algebra are also called as Boolean variables.

$$x + 0 = x$$

$$x.1 = x$$

$$x + 1 = 1$$

$$x.0 = 0$$

$$x + x = x$$

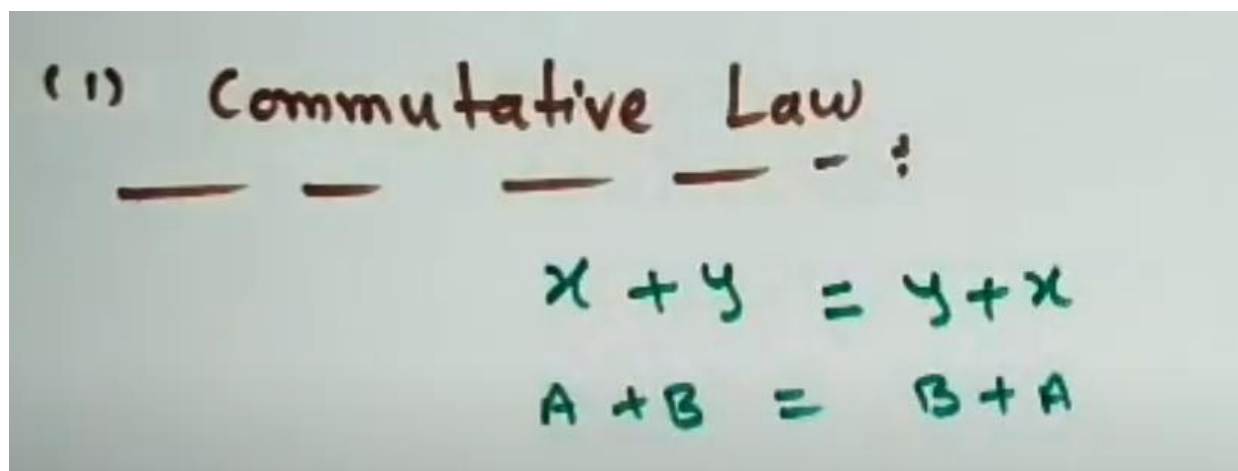
$$x.x = x$$

$$x + x' = 1$$

$$x.x' = 0$$

### Combinational circuits: Conversion of MUX and Decoders to logic gates:

- A combination circuit is one that has a "combination" of series and parallel paths for the electricity to flow. Its properties are a combination of the two. In this example, the parallel section of the circuit is like a sub-circuit and actually is part of an over-all series circuit.



## Associative Law

— — — —

$$x + (y + z) = (x + y) + z$$

$$A + (B + C) = (A + B) + C$$

## (3) Distributive Law

— — — — — ÷

$$x(y + z) = xy + xz$$

$$A(B + C) = AB + AC$$

- In computing and electronic systems, binary-coded decimal (BCD) is a class of binary encodings of decimal numbers where each digit is represented by a fixed number of bits, usually four or eight. Sometimes, special bit patterns are used for a sign or other indications (e.g. error or overflow).
- Binary Coded Decimal (BCD or "8421" BCD) numbers are made up using just 4 data bits (a nibble or half a byte) similar to the Hexadecimal numbers we saw in the binary tutorial, but unlike hexadecimal numbers that range in full from 0 through to F, BCD numbers only range from 0 to 9, with the binary number patterns of 1010 through to 1111 (A to F) being invalid inputs for this type of display and so are not used as shown below.

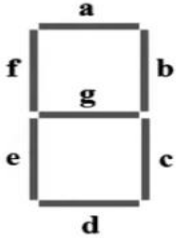
# MUX TO LOGIC GATES

1. NAND, NOR -Universal gates
  2. [“Universal Logic”](#)
  3. **MUX** and Decoders are called “Universal Logic”
4. now we will see how a 2:1 MUX can be used to create different logic gates.

## Design of 7 segment decoder with common anode display:

- The use of packed BCD allows two BCD digits to be stored within a single byte (8-bits) of data, allowing a single data byte to hold a BCD number in the range of 00 to 99. An example of the 4-bit BCD input ( 0100 ) representing the number “4” is given below

BCD to 7-segment decoder									
A	B	C	D	a	b	c	d	e	f
0	0	0	0	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0
0	0	1	0	1	1	0	1	1	0
0	0	1	1	1	1	1	1	0	0
0	1	0	0	0	1	1	0	0	1
0	1	0	1	1	0	1	1	0	1
0	1	1	0	1	0	1	1	1	1
0	1	1	1	1	1	1	0	0	0
1	0	0	0	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1



Date:	28 May 2020	Name:	Gagan M K
Course:	The Python Mega Course	USN:	4AL17EC032
Topic:	Application 5: Build a Desktop Database Application	Semester & Section:	6 <sup>th</sup> sem & 'A' sec

## AFTERNOON SESSION DETAILS

Image of session:

**Udemy** | The Python Mega Course: Build 10 Real World Applications

★ Leave a rating Your progress Share

# Solution

## Solution

```

1 def get_selected_row(event):
2     try:
3         global selected_tuple
4         index=list1.curselection()[0]
5         selected_tuple=list1.get(index)
6         e1.delete(0,END)
7         e1.insert(END,selected_tuple[1])
8         e2.delete(0,END)
9         e2.insert(END,selected_tuple[2])
10        e3.delete(0,END)
11        e3.insert(END,selected_tuple[3])
12        e4.delete(0,END)
13        e4.insert(END,selected_tuple[4])
14    except IndexError:
15        pass

```

**Course content**

- ☒ 187. Solution 1min
- ☐ 188. Creating a Standalone Executable Version of the Program 5min
- Section 24: Object Oriented Programming 0 / 8 | 1hr 15min
- Section 25: Python for Image and Video Processing with OpenCV 0 / 8 | 1hr 2min
- Section 26: Application 6: Build a Webcam Motion Detector 0 / 3 | 53min
- Section 27: Interactive Data Visualization with Bokeh 0 / 17 | 58min
- Section 28: Webscraping with Python Beautiful Soup 0 / 4 | 23min
- Section 29: Application 7: Scrape Real Estate Property Data from the Web

**About this course**

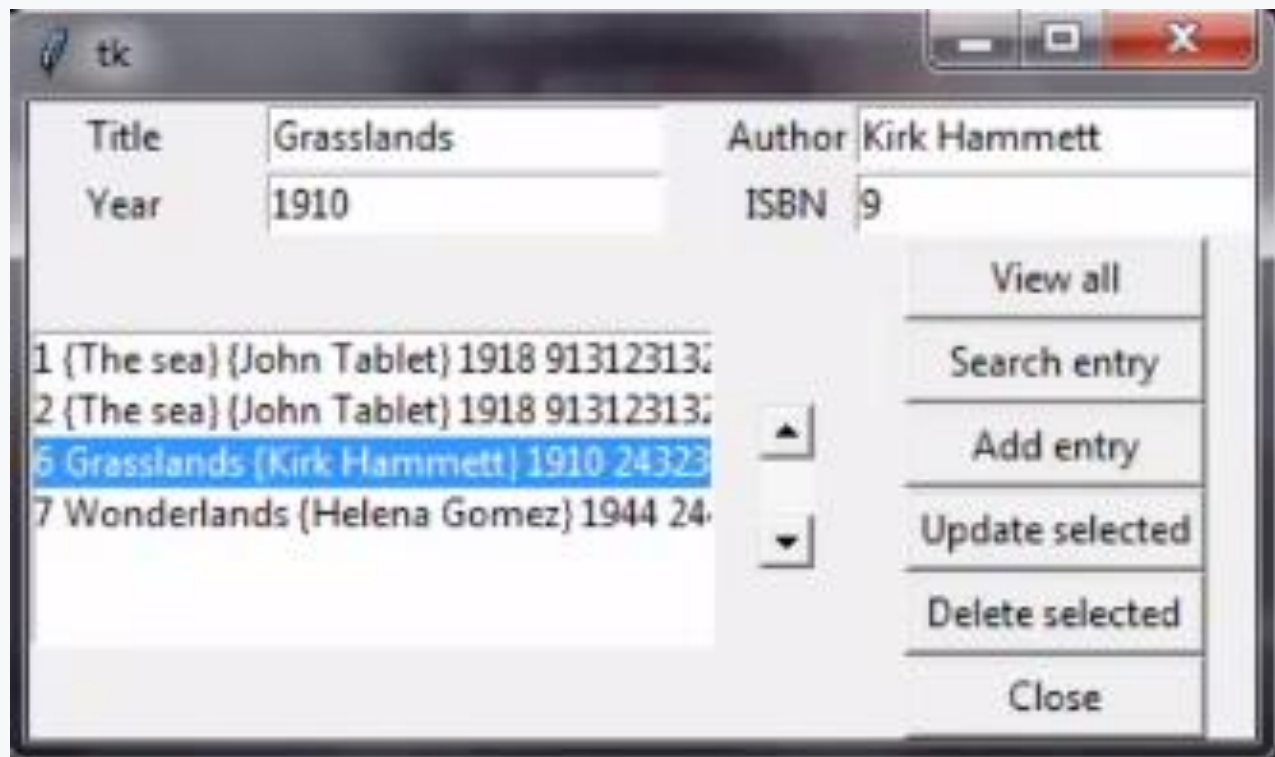
A complete Python course for both beginners and intermediates! Master Python 3 by making 10 amazing Python apps.

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

## Build a Desktop Database Application:

### CREATE DATABASE

- A Database is defined as a structured set of data. So, in SQL the very first step to store the data in a well structured manner is to create a database. The CREATE DATABASE statement is used to create a new database in SQL.



- The above picture is the window created using python.
- This Application was about creating Virtual book store by creating the window as shown above in the picture
- The program was divided into two parts Called frontend.py and backend.py and it creates a database in which all the books are stored in database.