# DAILY ASSESSMENT FORMAT

| Date: | 28/05/2020 | Name: | Prajwal Kamagethi Chakravarti P L |
|---|---|---|---|
| Course: | Python | USN: | 4AL17EC073 |
| Topic: | Application 5: Build a Desktop Database Application | Semester & Section: | 6 & B |
| Github Repository: | https://github.com/alvas-education-foundation/Prajwal-Kamagethi.git | | |

| FORENOON SESSION DETAILS |
|---|
| **Report – Report can be typed or hand written for up to two pages.** |



- **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.Python 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 necessary.Tkinter can be chosen.Then we must import all the modules of the file.**

  **from tkinter import**
  **window= Tk()**

- **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, column= 2)
```
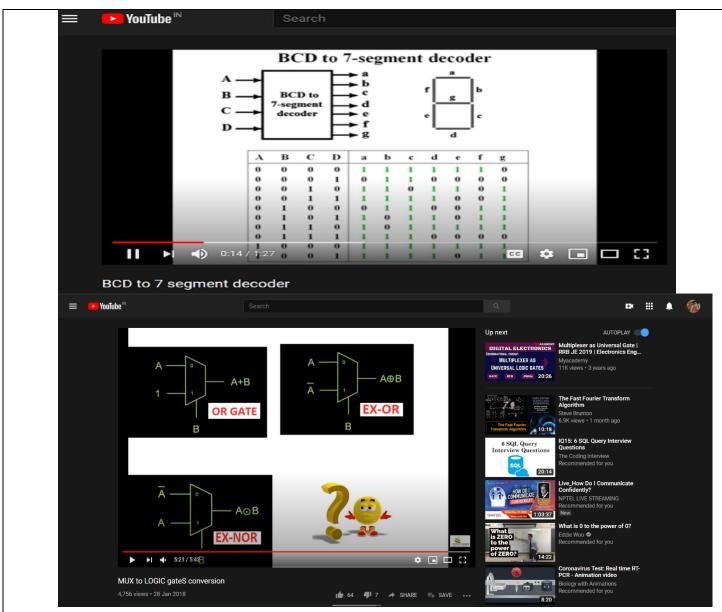
- **The button widget is very similar to the label widget. We create a variable and use the widget syntax to define what the button has to say.**

```
bt = Button (window, text="view all", widtg = 12)

bt.grid (column=3, row=2)
```

- **The grid function which is used to set the position of the button.We can set the scrolled text content by using the insert method. The syntax is pretty simple. We need to use txt.insert with the message as a parameter.Then we have to connect the frontend to backend using some of commands. The pro gram was divided into two parts Called frontend.py and backend.py and it creates a database in whic h all the books are stored in database.**

| Date: | 28-05-2020 | Name: | Prajwal Kamagethi Chakravarti P L |
|---|---|---|---|
| Course: | Digital signal processing | USN: | 4AL17EC073 |
| Topic: | 1 Boolean equations for digital circuits. Combinational circuits: Conversion of MUX and Decoders to logic gates. 2. Design of 7 segment decoder with common anode display | Semester & Section: | 6$^{TH}$ & B |
| Github Repository: | https://github.com/alvas-education-foundation/Prajwal-Kamagethi.git | | |

BCD to 7 segment decoder



MUX to LOGIC gateS conversion

4,756 views • 28 Jan 2018

**Report:**

## 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.**

$$x + 0 = x \qquad\qquad x.1 = x$$

$$x + 1 = 1 \qquad\qquad x.0 = 0$$

$$x + x = x \qquad\qquad x.x = x$$

$$x + x' = 1 \qquad\qquad x.x' = 0$$

- 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.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.

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