**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **28/05/2020** | **Name:** | **Namratha S Hipparagi** |
| **Course:** | **Logic design** | **USN:** | **4AL16EC040** |
| **Topic:** | **Boolean equations for digital circuits. Combinational circuits: Conversion of MUX and Decoders to logic gates.**  **design of 7 segment decoder with common anode display** | **Semester & Section:** | **8 A** |
| **Github Repository:** | **namrathahipparagi\_1** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| Report  Boolean algebra is a division of mathematics which deals with operations on logical values and incorporates binary variables. Boolean algebra is based upon a two-valued, or binary scheme. The two values may be expressed in many ways, such as true or false, 1 or 0, and "on" or "off".  Boolean laws:  Commutative Law  (a) A + B = B + A (b) A B = B A      Distributive Law  (a) A (B + C) = A B + A C (b) A + (B C) = (A + B) (A + C)  Associate Law  (a) (A + B) + C = A + (B + C) (b) (A B) C = A (B C)  Redundance Laws  (a) A + A B = A (b) A (A + B) = A  (a) 0 + A = A (b) 0 A = 0  (a) 1 + A = 1 (b) 1 A = A  (a) A+NOT A = 1 (b) A NOT A = 0  (a) A+NOT AB = A+B (b) A(NOT A+B) = AB   Identity Laws  (a) A + A = A (b) A A = A  (a) AB +ANOT B = A (b) (A+B)(A+NOT B) = A  Involution Law  (a) NOT A = A    De Morgan's Theorem  DeMorgan's Theorems are basically two sets of rules or laws developed from the Boolean expressions for AND, OR and NOT using two input variables, A and B.  (a) NOT A or B = NOT A NOT B   (Breaking the Overbar changes the OR to an AND) (b) NOT A = NOT A + NOT B   (Breaking the Overbar changes the AND to an OR) The Multiplexer The multiplexer is a combinational logic circuit designed to switch one of several input lines to a single common output line. In electronics, a multiplexer (or mux; spelled sometimes as multiplexor), also known as a data selector, is a device that selects between several analog or digital input signals and forwards it to a single output line.   Logic GatesA logic gate is an idealized or physical electronic device implementing a Boolean function, a logical operation performed on one or more binary inputs that produces a single binary output. The status of the input and output terminals can only be in one of the two binary conditions, either low (0) or high (1), represented by two different voltage levels, typically 0 volts for logic 0, and around 3 to 5 volts for logic 1, depending on the semiconductor technology used, Logic gates may have two or more inputs and, except in some special cases, they have a single output.. Logic gates also require a power supply. OR Gates |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **28/5/2020** | **Name:** | **Namratha S Hipparagi** | |
| **Course:** | **Python** | **USN:** | **4al16ec040** | |
| **Topic:** | **Application 5: Build a Desktop Database Application** | **Semester & Section:** | **8 A** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **REPORT**  **Front end:**  The term “front-end” refers to the user interface, while “back-end” means the server, application and database that work behind the scenes to deliver information to the user. The user enters a request through the interface.Front-end Web Development refers to building web interfaces, specifically the parts of the website that the user will interact with. When you’re browsing the web, everything you see, from images and headings to sliders and buttons is made using HTML, CSS and JavaScript, the main components to any website.  **Front End Frameworks and Libraries:**   * AngularJS: It is a continuously growing and expanding framework which provides better ways for developing web applications. It extends HTML attributes with Directives, and data is bound with HTML. It changes the static HTML to dynamic HTML. It is an open source project which can be freely used and changed by anyone. * jQuery: Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development. * SASS: It is the most reliable, mature and robust CSS extension language. It is used to extend the functionality of an existing CSS of a site including everything from variables, inheritance, and nesting with ease. * React.js:  ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook. * Some other libraries and frameworks are: Semantic-UI, Foundation, Materialize, Backbone.js, Express.js, Ember.js etc.   A back-end developer is a type of programmer who creates the logical back-end and core computational logic of a website, software or information system. The back end handles application logic, algorithms, database interaction and the processing of user requests. Back-end Development refers to the parts of the website that a user doesn’t see or directly interact with.  **Back end Languages:** The back end portion is built by using some languages which are discussed below:   * Object-oriented– This means the code is structured and modeled as a set of objects which can be manipulated, controlled and easily extended. * Basic– Java language is very basic and simple to master once you have understood the concept of OOP (Object Oriented Programming) * Platform independent– Java does not depend on any specific platform because, during compilation, it is compiled to an independent bytecode format which can be distributed and translated by JVM on any platform. * Robust– Java has good Memory management and Exception handling which makes it minimize error-prone code. * Multi-Threading– Java can run multiple tasks concurrently due to this feature which helps save a lot of memory.   **Connecting the Frontend to the Backend**  # Head of Tkinter application  master = Tk()  master.title("Network Automation")  # configuration for the labels and entry  Label(master, text="Device : ").grid(row=0)  Label(master, text="User ID : ").grid(row=1)  Label(master, text="Password : ").grid(row=2)  e1 = Entry(master)  e2 = Entry(master)  e3 = Entry(master, show='\*')  e1.grid(row=0, column=1)  e2.grid(row=1, column=1)  e3.grid(row=2, column=1, )  # configuration for the button  Button(master, text='Quit', command=master.destroy).grid(row=4, column=0, sticky=W, pady=4)  Button(master, text='Harden', command=show\_entry\_fields).grid(row=4, column=1, sticky=W, pady=4)  mainloop()  [python](https://stackoverflow.com/questions/tagged/python) [tkinter](https://stackoverflow.com/questions/tagged/tkinter" \o "show questions tagged 'tkinter') | | | |
|  | | | |