**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **9/June/2020** | **Name:** | **Prashantha naik** |
| **Course:** | **Kicad** | **USN:** | **4al17ec074** |
| **Topic:** | **1.Start a new project**  **2.Netlist and footprint association** | **Semester & Section:** | **6th b** |
| **GitHub Repository:** | **prashanth\_course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **Ki CAD is an open source software suite for Electronic Design Automation (EDA). The programs handle Schematic Capture, and PCB Layout with Gerber output. The suite runs on Windows, Linux and macOS and is licensed under GNU GPL v3.**  **The name of KiCad comes from the first letters of a company of Jean-Pierre Charras' friend "Ki" being combined with "Cad". But it now has no meaning other than being the name of the software suite.**   * **Learnt how modify the page settings** * **Learnt how place the all electrical component and connecting them by using wire.** * **Learnt how test circuit connection.** * **Learnt how to generate a netlist** * **Learnt how to assign a footprint value.** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **9/June/2020** | **Name:** | **Prashantha naik** | |
| **Course:** | **MySQL** | **USN:** | **4al17ec074** | |
| **Topic:** | **1.Outputting and processing data.**  **2.** **Dealing with variables**  **3.** **Inserting and using database data** | **Semester&Section:** | **6th b** | |
| **Git hub repository** | **prashanth\_couse** |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**  **MySQL LOOP**  **The loop \_label before the LOOP statement for using with the ITERATE and LEAVE statements.**  **If the value of x is greater than 10, the loop is terminated because of the LEAVEstatement.**  **If the value of the x is an odd number, the ITERATE ignores everything below it and starts a new loop iteration.**  **DELIMITER $$**  **CREATE PROCEDURE LoopDemo()**  **BEGIN**  **DECLARE x INT;**  **DECLARE str VARCHAR(255);**    **SET x = 1;**  **SET str = '';**    **loop\_label: LOOP**  **IF x > 10 THEN**  **LEAVE loop\_label;**  **END IF;**    **SET x = x + 1;**  **IF (x mod 2) THEN**  **ITERATE loop\_label;**  **ELSE**  **SET str = CONCAT(str,x,',');**  **END IF;**  **END LOOP;**  **SELECT str;**  **END$$**  **Dealing with variables**  **To create a user-defined variable, we use the format @variable\_name, where the variable\_name consists of alphanumeric characters. The maximum length of the user-defined variable is 64 characters**  **There are two ways to assign a value to a user-defined variable.**  **The first way is to use the SET statement as follows:**  **SET @variable\_name := value;**  **Inserting and using database data**  **The INSERT statement allows you to insert one or more rows into a table. The following illustrates the syntax of the INSERT statement:**  **INSERT INTO table(c1,c2,...)**  **VALUES (v1,v2,...);**  **In this syntax,**  **First, specify the table name and a list of comma-separated columns inside parentheses after the INSERT INTO clause.**  **Then, put a comma-separated list of values of the corresponding columns inside the parentheses following the VALUES keyword.**  **MySQL INSERT examples**  **CREATE TABLE IF NOT EXISTS tasks (**  **task\_id INT AUTO\_INCREMENT,**  **title VARCHAR(255) NOT NULL,**  **start\_date DATE,**  **due\_date DATE,**  **priority TINYINT NOT NULL DEFAULT 3,**  **description TEXT,**  **PRIMARY KEY (task\_id)**  **);** | | | |