**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **13/June/2020** | **Name:** | **Prashantha naik** |
| **Course:** | **Mysql** | **USN:** | **4al17ec074** |
| **Topic:** | **Variable, data structure** | **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.**  **Variables**  **Syntax:**  **<?php**  **// Assign the value "Hello!" to the variable "greeting"**  **$greeting = "Hello!";**  **// Assign the value 8 to the variable "month"**  **$month = 8;**  **// Assign the value 2019 to the variable "year"**  **$year = 2019;**  **?>**  **Strings**  **A string is a sequence of characters. It can be any text inside quotes (single or double):**  **$x = "Hello!";**  **$y = 'Hello!';**  **Integers**  **An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.**  **Rules for integers:**  **Integers must have at least one digit**  **Integers must not have a decimal point**  **Integers can be either positive or negative**  **$x = 5;**  **Floats**  **A float, or floating point number, is a number with a decimal point.**  **$x = 5.01;**  **Booleans**  **A Boolean represents two possible states: TRUE or FALSE. Booleans are often used in conditional testing.**  **$x = true;**  **$y = false;**  **Arrays**  **An array stores multiple values in one single variable.**  **$colors = array("Magenta", "Yellow", "Cyan");**  **NULL**  **Null is a special data type that can only have the value null. Variables can be declared with no value or emptied by setting the value to null. Also, if a variable is created without being assigned a value, it is automatically assigned null.**  **<?php**  **// Assign the value "Hello!" to greeting**  **$greeting = "Hello!";**  **// Empty the value greeting by setting it to null**  **$greeting = null;**  **?>**  **Classes and Objects**  **A class is a data structure useful for modeling things in the real world, and can contain properties and methods. Objects are instances a class, and are a convenient way to package values and functions specific to a class.**  **<?php**  **class Car {**  **function Car() {**  **$this->model = "Tesla";**  **}**  **}**  **// create an object**  **$Lightning = new Car();**  **// show object properties**  **echo $Lightning->model;**  **?>**  **Scope**  **It is important to know that unlike variables, constants ALWAYS have a global scope and can be accessed from any function in the script.**  **<?php**  **define("freeCodeCamp", "Learn to code and help nonprofits", false);**  **echo freeCodeCamp;**  **>?**  **// Output: Learn to code and help nonprofits**  **Operators**  **<?php**  **echo 1 <=> 1; // 0**  **echo 1 <=> 2; // -1**  **echo 2 <=> 1; // 1** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **13/June/2020** | **Name:** | **Prashantha naik** | |
| **Course:** | **MySQL** | **USN:** | **4al17ec074** | |
| **Topic:** | **Real life PHP Introduction** | **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.**  **encapsulation and polymorphism.**  **<?php**  **interface DirectoryIteratorInterface**  **{**  **/\*\***  **\* @return \Traversable|array**  **\*/**  **public function listDirs();**  **}**  **abstract class AbstractDirectoryIterator implements DirectoryIteratorInterface**  **{**  **protected $root;**  **public function \_\_construct($root)**  **{**  **$this->root = $root;**  **}**  **}**  **class LocalDirectoryIterator extends AbstractDirectoryIterator**  **{**  **public function listDirs()**  **{**  **// logic to get the current directory nodes and return them**  **}**  **}**  **class FtpDirectoryIterator extends AbstractDirectoryIterator**  **{**  **public function listDirs()**  **{**  **// logic to get the current directory nodes and return them**  **}**  **}**  **public function insert()**  **public function insert($table,$values,$rows = null)**  **{**  **if($this->tableExists($table))**  **{**  **$insert = 'INSERT INTO '.$table;**  **if($rows != null)**  **{**  **$insert .= ' ('.$rows.')';**  **}**    **for($i = 0; $i < count($values); $i++)**  **{**  **if(is\_string($values[$i]))**  **$values[$i] = '"'.$values[$i].'"';**  **}**  **$values = implode(',',$values);**  **$insert .= ' VALUES ('.$values.')';**  **$ins = @mysql\_query($insert);**  **if($ins)**  **{**  **return true;**  **}**  **else**  **{**  **return false;**  **}**  **}**  **}**  **public function delete()**  **public function delete($table,$where = null)**  **{**  **if($this->tableExists($table))**  **{**  **if($where == null)**  **{**  **$delete = 'DELETE '.$table;**  **}**  **else**  **{**  **$delete = 'DELETE FROM '.$table.' WHERE '.$where;**  **}**  **$del = @mysql\_query($delete);**    **if($del)**  **{**  **return true;**  **}**  **else**  **{**  **return false;**  **}**  **}**  **else**  **{**  **return false;**  **}**  **}** | | | |