Practical List OSTP

1. Write a program to print “Hello World”.

<?php

echo "Hello World";

?>

1. Write a program to define Static, global and local variable.

Static

<?php

function increment() {

static $count = 0; // Static variable

$count++;

echo "Count: $count<br>";

}

increment(); // Output: Count: 1

increment(); // Output: Count: 2

increment(); // Output: Count: 3

?>

Global

<?php

$globalVar = "I'm a global variable";

function printGlobal() {

global $globalVar; // Access global variable inside the function

echo $globalVar;

}

printGlobal(); // Output: I'm a global variable

?>

Local

<?php

function myFunction() {

$localVar = "I'm a local variable"; // Local variable

echo $localVar;

}

myFunction(); // Output: I'm a local variable

// echo $localVar; // This will cause an error as $localVar is not defined in the global scope

?>

1. Write a program to create a form using GET & POST method retrieve the value using $\_GET and $\_POST in second page.

Form.php

<!DOCTYPE html>

<html>

<head>

<title>Form Page 1</title>

</head>

<body>

<h1>Form Page 1</h1>

<form action="process.php" method="get">

Name: <input type="text" name="name"><br>

Email: <input type="text" name="email"><br>

<input type="submit" value="Submit (GET)">

</form>

<br>

<form action="process.php" method="post">

Name: <input type="text" name="name"><br>

Email: <input type="text" name="email"><br>

<input type="submit" value="Submit (POST)">

</form>

</body>

</html>

Process.php

<!DOCTYPE html>

<html>

<head>

<title>Form Processing</title>

</head>

<body>

<h1>Form Processing</h1>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "GET") {

$name = $\_GET["name"];

$email = $\_GET["email"];

echo "Using GET Method:<br>";

echo "Name: $name<br>";

echo "Email: $email<br>";

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$name = $\_POST["name"];

$email = $\_POST["email"];

echo "Using POST Method:<br>";

echo "Name: $name<br>";

echo "Email: $email<br>";

}

?>

</body>

</html>

1. Write a program to perform arithmetic operators.

<?php

$number1 = 10;

$number2 = 5;

// Addition

$sum = $number1 + $number2;

echo "Addition: " . $sum . "<br>";

// Subtraction

$difference = $number1 - $number2;

echo "Subtraction: " . $difference . "<br>";

// Multiplication

$product = $number1 \* $number2;

echo "Multiplication: " . $product . "<br>";

// Division

$quotient = $number1 / $number2;

echo "Division: " . $quotient . "<br>";

// Modulo (Remainder after division)

$remainder = $number1 % $number2;

echo "Modulo (Remainder): " . $remainder . "<br>";

// Increment

$number1++;

echo "Increment: " . $number1 . "<br>";

// Decrement

$number2--;

echo "Decrement: " . $number2 . "<br>";

?>

1. Write a program to perform conditional structure.

<?php

$age = 25;

if ($age < 18) {

echo "You are a minor.";

} else if ($age >= 18 && $age < 60) {

echo "You are an adult.";

} else {

echo "You are a senior citizen.";

}

?>

1. Write a program to perform looping structure.

For loop

<?php

for ($i = 1; $i <= 5; $i++) {

echo "$i ";

}

?>

While loop

<?php

$counter = 1;

while ($counter <= 5) {

echo "$counter ";

$counter++;

}

?>

Do-While loop

<?php

$counter = 1;

do {

echo "$counter ";

$counter++;

} while ($counter <= 5);

?>

Foreach loop(for arrays)

<?php

$colors = array("Red", "Green", "Blue", "Yellow");

foreach ($colors as $color) {

echo "$color ";

}

?>

**Nested Loops (Example: Multiplication Table)**

<?php

for ($i = 1; $i <= 5; $i++) {

for ($j = 1; $j <= 5; $j++) {

echo $i \* $j . " ";

}

echo "<br>";

}

?>

1. Write a program to perform all the array built in function.

<?php

// Initialize an array

$numbers = array(10, 5, 8, 3, 2, 7, 5, 1);

// Displaying the original array

echo "Original Array: ";

print\_r($numbers);

echo "<br>";

// 1. Sorting Functions

echo "\n\n1. Sorting Functions:\n";

sort($numbers); // Sorts the array in ascending order

echo "Sorted (Ascending): ";

print\_r($numbers);

echo "<br>";

rsort($numbers); // Sorts the array in descending order

echo "Sorted (Descending): ";

print\_r($numbers);

echo "<br>";

asort($numbers); // Sorts the array by values, maintaining index association

echo "Sorted by Value (Ascending): ";

print\_r($numbers);

echo "<br>";

// 2. Searching Functions

echo "\n2. Searching Functions:\n";

echo "In\_array (Value exists in array): ";

echo in\_array(8, $numbers) ? "Yes" : "No";

echo "\nArray\_search (Searches value and returns corresponding key): ";

$key = array\_search(5, $numbers);

echo $key !== false ? $key : "Not Found";

echo "<br>";

// 3. Array Manipulation Functions

echo "\n\n3. Array Manipulation Functions:\n";

array\_push($numbers, 6); // Adds an element to the end of the array

echo "After array\_push (Add 6 to end): ";

print\_r($numbers);

echo "<br>";

array\_pop($numbers); // Removes and returns the last element of the array

echo "After array\_pop (Remove last element): ";

print\_r($numbers);

echo "<br>";

array\_shift($numbers); // Removes and returns the first element of the array

echo "After array\_shift (Remove first element): ";

print\_r($numbers);

echo "<br>";

array\_unshift($numbers, 0); // Adds elements to the beginning of the array

echo "After array\_unshift (Add 0 to beginning): ";

print\_r($numbers);

echo "<br>";

// 4. Array Functions for Counting and Sizing

echo "\n4. Array Functions for Counting and Sizing:\n";

echo "Count (Number of elements in array): " . count($numbers);

echo "\nMax (Maximum value in array): " . max($numbers);

echo "\nMin (Minimum value in array): " . min($numbers);

echo "<br>";

// 5. Array Functions for Extracting and Copying

echo "\n\n5. Array Functions for Extracting and Copying:\n";

$subset = array\_slice($numbers, 2, 3); // Extract a portion of the array

echo "Subset of array: ";

print\_r($subset);

echo "<br>";

$copy = array\_splice($numbers, 2, 3); // Extract and remove a portion of the array

echo "Copied and Removed from array: ";

print\_r($copy);

echo "<br>";

// 6. Array Functions for Merging and Combining

echo "\n6. Array Functions for Merging and Combining:\n";

$moreNumbers = array(11, 12, 13);

$merged = array\_merge($numbers, $moreNumbers); // Merge two arrays

echo "Merged Array: ";

print\_r($merged);

$combined = array\_combine($numbers, $moreNumbers); // Combine two arrays into one using keys from one and values from another

echo "Combined Array: ";

print\_r($combined);

echo "<br>";

// 7. Array Functions for Filtering and Reducing

echo "\n7. Array Functions for Filtering and Reducing:\n";

$filtered = array\_filter($numbers, function($value) {

return $value % 2 == 0; // Filter even numbers

});

echo "Filtered (Even Numbers): ";

print\_r($filtered);

$sum = array\_reduce($numbers, function($carry, $item) {

return $carry + $item; // Calculate sum of array

});

echo "Sum of Array: " . $sum;

echo "<br>";

// 8. Array Functions for Transforming

echo "\n\n8. Array Functions for Transforming:\n";

$mapped = array\_map(function($value) {

return $value \* 2; // Multiply each element by 2

}, $numbers);

echo "Mapped (Doubled): ";

print\_r($mapped);

$flipped = array\_flip($numbers); // Swap keys and values

echo "Flipped Array: ";

print\_r($flipped);

?>

1. Write a program to perform all the string built in functions.

<?php

$string = "Hello, World!";

// 1. String Length

echo "1. String Length: " . strlen($string) . "<br>";

// 2. String to Lowercase and Uppercase

echo "2. To Lowercase: " . strtolower($string) . "<br>";

echo " To Uppercase: " . strtoupper($string) . "<br>";

// 3. Substring

echo "3. Substring (5, 5): " . substr($string, 5, 5) . "<br>";

// 4. String Replace

$replacedString = str\_replace("Hello", "Hi", $string);

echo "4. String Replace (Hello with Hi): " . $replacedString . "<br>";

// 5. String Position (Finding a Substring)

$position = strpos($string, "World");

echo "5. Position of 'World': " . $position . "<br>";

// 6. String Trim

$trimmedString = trim(" Some spaces before and after ");

echo "6. Trimmed String: '" . $trimmedString . "'<br>";

// 7. String Reverse

$reversedString = strrev($string);

echo "7. Reversed String: " . $reversedString . "<br>";

// 8. String Comparison

$string1 = "Hello";

$string2 = "hello";

if (strcasecmp($string1, $string2) == 0) {

echo "8. Strings are equal (case-insensitive)<br>";

} else {

echo "8. Strings are not equal (case-insensitive)<br>";

}

// 9. String Repeat

$repeatedString = str\_repeat("Hello, ", 3);

echo "9. Repeated String: " . $repeatedString . "<br>";

// 10. String Length of Words

$words = explode(" ", $string);

foreach ($words as $word) {

echo "Length of '$word': " . strlen($word) . "<br>";

}

// 11. String Character Count

$charCount = count\_chars($string, 1);

echo "11. Character Count: ";

print\_r($charCount);

echo "<br>";

// 12. String Comparison

$comparisonResult = strcmp("abc", "abc");

echo "12. String Comparison: " . $comparisonResult . "<br>";

?>

1. Pattern:

1

12

123

1234

<?php

$rows = 4;

for ($i = 1; $i <= $rows; $i++) {

for ($j = 1; $j <= $i; $j++) {

echo $j;

}

echo "<br>";

}

?>

1. Pattern :

1

22

333

4444

<?php

$rows = 4;

for ($i = 1; $i <= $rows; $i++) {

for ($j = 1; $j <= $i; $j++) {

echo $i;

}

echo "<br>";

}

?>

1. Pattern:

1234

123

12

1

<?php

$rows = 4;

for ($i = $rows; $i >= 1; $i--) {

for ($j = 1; $j <= $i; $j++) {

echo $j;

}

echo "<br>";

}

?>

1. Pattern:

4444

333

22

1

<?php

$rows = 4;

for ($i = $rows; $i >= 1; $i--) {

for ($j = 1; $j <= $i; $j++) {

echo $i;

}

echo "<br>";

}

?>

1. Write a program for cookies in PHP.

<?php

// Set a cookie with a name, value, and expiration time (in seconds)

setcookie("username", "john\_doe", time() + 3600, "/"); // Expires in 1 hour

// Retrieve and display the value of the cookie

if(isset($\_COOKIE["username"])) {

echo "Welcome, " . $\_COOKIE["username"] . "!";

} else {

echo "Cookie not set.";

}

// Delete a cookie by setting its expiration time to the past

setcookie("username", "", time() - 3600, "/"); // Delete the cookie

// Check if the cookie is deleted

if(isset($\_COOKIE["username"])) {

echo "<br>Cookie still exists: " . $\_COOKIE["username"];

} else {

echo "<br>Cookie deleted.";

}

?>

1. Write a program for session in PHP.

Index.php

<?php

session\_start(); // Start a new session or resume an existing session

$\_SESSION['username'] = 'john\_doe'; // Set a session variable

echo "Session variable 'username' set.<br>";

echo "<a href='page2.php'>Go to Page 2</a>";

?>

Page2.php

<?php

session\_start(); // Start or resume the session

if (isset($\_SESSION['username'])) {

echo "Welcome, " . $\_SESSION['username'] . "!<br>";

} else {

echo "Session variable 'username' not set.<br>";

}

echo "<a href='logout.php'>Logout</a>";

?>

Logout.php

<?php

session\_start(); // Start or resume the session

session\_unset(); // Unset all session variables

session\_destroy(); // Destroy the session

echo "Session destroyed. You are now logged out.<br>";

echo "<a href='index.php'>Go to Home Page</a>";

?>

1. Write a program to perform file uploading in PHP.

Upload.php

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>File Upload</title>

</head>

<body>

<h1>Upload a File</h1>

<form action="upload.php" method="post" enctype="multipart/form-data">

<input type="file" name="fileToUpload" id="fileToUpload">

<input type="submit" value="Upload File" name="submit">

</form>

</body>

</html>

Upload.php(PHP script for handling file upload)

<?php

$target\_dir = "uploads/"; // Directory where uploaded files will be stored

$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]); // Path of the uploaded file

$uploadOk = 1; // Flag to indicate if file upload was successful

$imageFileType = strtolower(pathinfo($target\_file,PATHINFO\_EXTENSION)); // Get the file extension

// Check if the file is an actual file or fake

if(isset($\_POST["submit"])) {

$check = getimagesize($\_FILES["fileToUpload"]["tmp\_name"]);

if($check !== false) {

echo "File is an image - " . $check["mime"] . ".<br>";

$uploadOk = 1;

} else {

echo "File is not an image.<br>";

$uploadOk = 0;

}

}

// Check if file already exists

if (file\_exists($target\_file)) {

echo "Sorry, file already exists.<br>";

$uploadOk = 0;

}

// Check file size (max 5MB)

if ($\_FILES["fileToUpload"]["size"] > 5000000) {

echo "Sorry, your file is too large. Max size allowed is 5MB.<br>";

$uploadOk = 0;

}

// Allow certain file formats (in this example, only JPG, JPEG, PNG, and GIF are allowed)

$allowedExtensions = array("jpg", "jpeg", "png", "gif");

if(!in\_array($imageFileType, $allowedExtensions)) {

echo "Sorry, only JPG, JPEG, PNG, and GIF files are allowed.<br>";

$uploadOk = 0;

}

// Check if $uploadOk is set to 0 by an error

if ($uploadOk == 0) {

echo "Sorry, your file was not uploaded.<br>";

} else {

// If everything is ok, try to upload the file

if (move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"], $target\_file)) {

echo "The file ". htmlspecialchars( basename( $\_FILES["fileToUpload"]["name"])). " has been uploaded.<br>";

} else {

echo "Sorry, there was an error uploading your file.<br>";

}

}

?>

1. Write a program to send a mail threw PHP script.

<?php

$to = "recipient@example.com"; // Recipient's email address

$subject = "Test Email"; // Subject of the email

$message = "This is a test email sent from a PHP script."; // Body of the email

$headers = "From: sender@example.com"; // Sender's email address

// Send the email

$mailSent = mail($to, $subject, $message, $headers);

if ($mailSent) {

echo "Email sent successfully!";

} else {

echo "Failed to send email. Please check your configuration.";

}

?>

1. Write a program to perform AJAX functionality in PHP.

**index.html** (HTML file with AJAX functionality):

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>AJAX Example</title>

<script>

function loadContent() {

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("content").innerHTML = this.responseText;

}

};

xhttp.open("GET", "ajax.php", true);

xhttp.send();

}

</script>

</head>

<body>

<h1>AJAX Example</h1>

<button onclick="loadContent()">Load Content</button>

<div id="content"></div>

</body>

</html>

**ajax.php** (PHP script to handle AJAX request):

<?php

echo "This is dynamically loaded content from ajax.php";

?>

1. Write a program to perform AJAX functionality using PHP and mysql.

**index.html** (HTML file with AJAX functionality):

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>AJAX Example with MySQL</title>

<script>

function loadContent() {

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("content").innerHTML = this.responseText;

}

};

xhttp.open("GET", "ajax.php", true);

xhttp.send();

}

</script>

</head>

<body>

<h1>AJAX Example with MySQL</h1>

<button onclick="loadContent()">Load Content</button>

<div id="content"></div>

</body>

</html>

**ajax.php** (PHP script to handle AJAX request and interact with MySQL):

<?php

$servername = "localhost";

$username = "username"; // Replace with your MySQL username

$password = "password"; // Replace with your MySQL password

$dbname = "test\_db"; // Replace with your MySQL database name

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "SELECT \* FROM users";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

echo "<ul>";

while($row = $result->fetch\_assoc()) {

echo "<li>" . $row["username"] . "</li>";

}

echo "</ul>";

} else {

echo "0 results";

}

$conn->close();

?>

1. Write program to perform JSON function json\_encode and json\_decode.

<?php

// Create a PHP array

$phpArray = array(

'name' => 'John Doe',

'age' => 30,

'city' => 'New York'

);

// Encode the PHP array into a JSON string

$jsonString = json\_encode($phpArray);

// Print the JSON string

echo "JSON String: " . $jsonString . "<br>";

// Decode the JSON string back into a PHP array

$decodedArray = json\_decode($jsonString, true);

// Print the decoded PHP array

echo "Decoded PHP Array: ";

print\_r($decodedArray);

?>

1. Create database paruluniversity create a table MCA and put the record in mca table with field stud\_id,stud\_enroll,stud\_name,stud\_gender,stud\_mob,stud\_gender,stud\_dob.

<?php

$servername = "localhost";

$username = "root"; // Replace with your MySQL username

$password = ""; // Replace with your MySQL password

// Create connection

$conn = new mysqli($servername, $username, $password);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Create the database

$sqlCreateDB = "CREATE DATABASE IF NOT EXISTS paruluniversity";

if ($conn->query($sqlCreateDB) === TRUE) {

echo "Database created successfully<br>";

} else {

echo "Error creating database: " . $conn->error . "<br>";

}

// Select the database

$conn->select\_db("paruluniversity");

// Create the MCA table

$sqlCreateTable = "CREATE TABLE IF NOT EXISTS MCA (

stud\_id INT AUTO\_INCREMENT PRIMARY KEY,

stud\_enroll VARCHAR(20) UNIQUE,

stud\_name VARCHAR(100),

stud\_gender ENUM('Male', 'Female', 'Other'),

stud\_mob VARCHAR(10),

stud\_dob DATE

)";

if ($conn->query($sqlCreateTable) === TRUE) {

echo "Table MCA created successfully<br>";

} else {

echo "Error creating table: " . $conn->error . "<br>";

}

// Insert a record into the MCA table

$sqlInsertRecord = "INSERT INTO MCA (stud\_enroll, stud\_name, stud\_gender, stud\_mob, stud\_dob)

VALUES ('MCA2021001', 'John Doe', 'Male', '9876543210', '1990-12-25')";

if ($conn->query($sqlInsertRecord) === TRUE) {

echo "Record inserted successfully<br>";

} else {

echo "Error inserting record: " . $conn->error . "<br>";

}

// Close connection

$conn->close();

?>

1. Write a program to display practical no 20 table records threw PHP script in tabular format.

<?php

$servername = "localhost";

$username = "root"; // Replace with your MySQL username

$password = ""; // Replace with your MySQL password

$dbname = "paruluniversity";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Retrieve records from the MCA table

$sqlSelect = "SELECT \* FROM MCA";

$result = $conn->query($sqlSelect);

if ($result->num\_rows > 0) {

echo "<table border='1'>

<tr>

<th>Student ID</th>

<th>Enrollment</th>

<th>Name</th>

<th>Gender</th>

<th>Mobile</th>

<th>Date of Birth</th>

</tr>";

while($row = $result->fetch\_assoc()) {

echo "<tr>

<td>". $row["stud\_id"]. "</td>

<td>". $row["stud\_enroll"]. "</td>

<td>". $row["stud\_name"]. "</td>

<td>". $row["stud\_gender"]. "</td>

<td>". $row["stud\_mob"]. "</td>

<td>". $row["stud\_dob"]. "</td>

</tr>";

}

echo "</table>";

} else {

echo "0 results";

}

// Close connection

$conn->close();

?>

1. Write a PHP program to create a form and put the records on the table.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Student Form</title>

</head>

<body>

<h1>Student Information Form</h1>

<form action="store\_record.php" method="post">

<label for="stud\_id">Student ID:</label>

<input type="text" id="stud\_id" name="stud\_id"><br><br>

<label for="stud\_enroll">Enrollment:</label>

<input type="text" id="stud\_enroll" name="stud\_enroll"><br><br>

<label for="stud\_name">Name:</label>

<input type="text" id="stud\_name" name="stud\_name"><br><br>

<label for="stud\_gender">Gender:</label>

<input type="radio" id="male" name="stud\_gender" value="Male">

<label for="male">Male</label>

<input type="radio" id="female" name="stud\_gender" value="Female">

<label for="female">Female</label>

<input type="radio" id="other" name="stud\_gender" value="Other">

<label for="other">Other</label><br><br>

<label for="stud\_mob">Mobile:</label>

<input type="text" id="stud\_mob" name="stud\_mob"><br><br>

<label for="stud\_dob">Date of Birth:</label>

<input type="date" id="stud\_dob" name="stud\_dob"><br><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

1. Write a PHP script to edit the records in database.

**edit\_record.php** (PHP script for editing records):

<?php

$servername = "localhost";

$username = "root"; // Replace with your MySQL username

$password = ""; // Replace with your MySQL password

$dbname = "paruluniversity";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$stud\_id = $\_POST['stud\_id'];

$stud\_enroll = $\_POST['stud\_enroll'];

$stud\_name = $\_POST['stud\_name'];

$stud\_gender = $\_POST['stud\_gender'];

$stud\_mob = $\_POST['stud\_mob'];

$stud\_dob = $\_POST['stud\_dob'];

// Update record in MCA table

$sqlUpdate = "UPDATE MCA SET stud\_enroll='$stud\_enroll', stud\_name='$stud\_name', stud\_gender='$stud\_gender', stud\_mob='$stud\_mob', stud\_dob='$stud\_dob' WHERE stud\_id=$stud\_id";

if ($conn->query($sqlUpdate) === TRUE) {

echo "Record updated successfully";

} else {

echo "Error updating record: " . $conn->error;

}

}

$conn->close();

?>

**edit\_form.html** (HTML form to select a record for editing):

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Edit Student Record</title>

</head>

<body>

<h1>Edit Student Record</h1>

<form action="edit\_record.php" method="post">

<label for="stud\_id">Student ID:</label>

<input type="text" id="stud\_id" name="stud\_id"><br><br>

<input type="submit" value="Edit Record">

</form>

</body>

</html>

**edit\_form.php** (PHP script to display the form):

<?php

include 'edit\_form.html';

?>

1. Write a PHP script to delete records in the database.

**delete\_record.php** (PHP script for deleting records):

<?php

$servername = "localhost";

$username = "root"; // Replace with your MySQL username

$password = ""; // Replace with your MySQL password

$dbname = "paruluniversity";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$stud\_id = $\_POST['stud\_id'];

// Delete record from MCA table

$sqlDelete = "DELETE FROM MCA WHERE stud\_id=$stud\_id";

if ($conn->query($sqlDelete) === TRUE) {

echo "Record deleted successfully";

} else {

echo "Error deleting record: " . $conn->error;

}

}

$conn->close();

?>

**delete\_form.html** (HTML form to select a record for deletion):

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Delete Student Record</title>

</head>

<body>

<h1>Delete Student Record</h1>

<form action="delete\_record.php" method="post">

<label for="stud\_id">Student ID:</label>

<input type="text" id="stud\_id" name="stud\_id"><br><br>

<input type="submit" value="Delete Record">

</form>

</body>

</html>

**delete\_form.php** (PHP script to display the form):

<?php

include 'delete\_form.html';

?>

1. Write a program to perform JQuery events like : Click, dbclick, keypress, keydown, keyup, submit,change, focus, blur, load, resize, scroll, unlode.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>jQuery Events</title>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script>

$(document).ready(function(){

// Click Event

$("#clickButton").click(function(){

alert("Button Clicked");

});

// Double Click Event

$("#doubleClickButton").dblclick(function(){

alert("Button Double Clicked");

});

// Key Press Event

$("#keypressInput").keypress(function(){

alert("Key Pressed");

});

// Key Down Event

$("#keydownInput").keydown(function(){

alert("Key Down");

});

// Key Up Event

$("#keyupInput").keyup(function(){

alert("Key Up");

});

// Submit Event

$("#submitForm").submit(function(event){

event.preventDefault();

alert("Form Submitted");

});

// Change Event

$("#changeSelect").change(function(){

alert("Select Value Changed");

});

// Focus Event

$("#focusInput").focus(function(){

alert("Input Focused");

});

// Blur Event

$("#blurInput").blur(function(){

alert("Input Blurred");

});

// Load Event

$(window).load(function(){

alert("Page Loaded");

});

// Resize Event

$(window).resize(function(){

alert("Window Resized");

});

// Scroll Event

$(window).scroll(function(){

alert("Window Scrolled");

});

// Unload Event

$(window).on('beforeunload', function(){

return "Are you sure you want to leave?";

});

});

</script>

</head>

<body>

<button id="clickButton">Click Me</button>

<button id="doubleClickButton">Double Click Me</button>

<input type="text" id="keypressInput" placeholder="Press a key">

<input type="text" id="keydownInput" placeholder="Press a key">

<input type="text" id="keyupInput" placeholder="Release a key">

<form id="submitForm">

<input type="submit" value="Submit Form">

</form>

<select id="changeSelect">

<option value="option1">Option 1</option>

<option value="option2">Option 2</option>

<option value="option3">Option 3</option>

</select>

<input type="text" id="focusInput" placeholder="Click to focus">

<input type="text" id="blurInput" placeholder="Click outside to blur">

</body>

</html>

1. Write a program to perform JQuery effects like : hide show, fade, slide.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>jQuery Effects</title>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script>

$(document).ready(function(){

// Hide and Show Effect

$("#hideButton").click(function(){

$("#hideShowDiv").hide(1000); // Hide with animation (1000 milliseconds)

});

$("#showButton").click(function(){

$("#hideShowDiv").show(1000); // Show with animation (1000 milliseconds)

});

// Fade Effect

$("#fadeButton").click(function(){

$("#fadeDiv").fadeToggle(1000); // Toggle fade in/out with animation (1000 milliseconds)

});

// Slide Effect

$("#slideDownButton").click(function(){

$("#slideDiv").slideDown(1000); // Slide down with animation (1000 milliseconds)

});

$("#slideUpButton").click(function(){

$("#slideDiv").slideUp(1000); // Slide up with animation (1000 milliseconds)

});

});

</script>

</head>

<body>

<h1>jQuery Effects</h1>

<button id="hideButton">Hide</button>

<button id="showButton">Show</button>

<div id="hideShowDiv" style="width: 100px; height: 100px; background-color: lightblue;"></div>

<br><br>

<button id="fadeButton">Fade</button>

<div id="fadeDiv" style="width: 100px; height: 100px; background-color: lightgreen;"></div>

<br><br>

<button id="slideDownButton">Slide Down</button>

<button id="slideUpButton">Slide Up</button>

<div id="slideDiv" style="width: 100px; height: 100px; background-color: lightcoral;"></div>

</body>

</html>

1. Write a program to create a class and object.

<?php

class Person {

// Properties (attributes)

public $name;

public $age;

// Constructor

public function \_\_construct($name, $age) {

$this->name = $name;

$this->age = $age;

}

// Method

public function greet() {

return "Hello, my name is {$this->name} and I am {$this->age} years old.";

}

}

// Create an object of the Person class

$person1 = new Person("John Doe", 30);

// Access properties and call method

echo $person1->name . "<br>"; // Output: John Doe

echo $person1->age . "<br>"; // Output: 30

echo $person1->greet() . "<br>"; // Output: Hello, my name is John Doe and I am 30 years old.

?>

1. Write a program to perform constructor and destructor in class.

<?php

class MyClass {

// Constructor

public function \_\_construct() {

echo "Constructor called<br>";

}

// Destructor

public function \_\_destruct() {

echo "Destructor called<br>";

}

public function sayHello() {

echo "Hello!<br>";

}

}

// Create an object of MyClass

$obj = new MyClass();

// Call a method of MyClass

$obj->sayHello();

// Object is automatically destroyed when it goes out of scope

?>

1. Write a program to perform inheritance.

<?php

class Animal {

public $name;

public function \_\_construct($name) {

$this->name = $name;

}

public function makeSound() {

return "Generic animal sound";

}

}

class Dog extends Animal {

public function makeSound() {

return "Bark";

}

public function wagTail() {

return "{$this->name} is wagging its tail";

}

}

class Cat extends Animal {

public function makeSound() {

return "Meow";

}

public function purr() {

return "{$this->name} is purring";

}

}

// Create objects

$dog = new Dog("Rex");

$cat = new Cat("Whiskers");

// Call methods

echo $dog->makeSound() . "<br>"; // Output: Bark

echo $dog->wagTail() . "<br>"; // Output: Rex is wagging its tail

echo $cat->makeSound() . "<br>"; // Output: Meow

echo $cat->purr() . "<br>"; // Output: Whiskers is purring

?>

1. Write a program to perform scope resolution operator in class.

<?php

class MyClass {

const MY\_CONSTANT = "Hello, I'm a constant";

public static $myStaticVariable = "I'm a static variable";

public static function myStaticMethod() {

return "I'm a static method";

}

public function myMethod() {

return self::MY\_CONSTANT . "<br>" .

self::$myStaticVariable . "<br>" .

self::myStaticMethod();

}

}

// Accessing static properties and methods using scope resolution operator

echo MyClass::MY\_CONSTANT . "<br>"; // Output: Hello, I'm a constant

echo MyClass::$myStaticVariable . "<br>"; // Output: I'm a static variable

echo MyClass::myStaticMethod() . "<br>"; // Output: I'm a static method

// Creating an object of MyClass

$obj = new MyClass();

// Accessing constants, static variables, and calling static methods using object

echo $obj->myMethod() . "<br>";

?>

1. Write a program to perform Mysql Database handling with oop (insert, update, select, delete).

<?php

class Database {

private $servername;

private $username;

private $password;

private $dbname;

private $conn;

public function \_\_construct($servername, $username, $password, $dbname) {

$this->servername = $servername;

$this->username = $username;

$this->password = $password;

$this->dbname = $dbname;

// Create connection

$this->conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($this->conn->connect\_error) {

die("Connection failed: " . $this->conn->connect\_error);

}

}

public function insertRecord($name, $email) {

$sql = "INSERT INTO users (name, email) VALUES ('$name', '$email')";

if ($this->conn->query($sql) === TRUE) {

return "Record inserted successfully";

} else {

return "Error: " . $sql . "<br>" . $this->conn->error;

}

}

public function updateRecord($id, $name, $email) {

$sql = "UPDATE users SET name='$name', email='$email' WHERE id=$id";

if ($this->conn->query($sql) === TRUE) {

return "Record updated successfully";

} else {

return "Error updating record: " . $this->conn->error;

}

}

public function selectAllRecords() {

$result = $this->conn->query("SELECT \* FROM users");

if ($result->num\_rows > 0) {

$records = array();

while ($row = $result->fetch\_assoc()) {

$records[] = $row;

}

return $records;

} else {

return "No records found";

}

}

public function deleteRecord($id) {

$sql = "DELETE FROM users WHERE id=$id";

if ($this->conn->query($sql) === TRUE) {

return "Record deleted successfully";

} else {

return "Error deleting record: " . $this->conn->error;

}

}

public function closeConnection() {

$this->conn->close();

}

}

// Usage example

$db = new Database("localhost", "username", "password", "database\_name");

// Insert a record

$insertResult = $db->insertRecord("John Doe", "john@example.com");

echo $insertResult . "<br>";

// Update a record

$updateResult = $db->updateRecord(1, "Jane Doe", "jane@example.com");

echo $updateResult . "<br>";

// Select all records

$records = $db->selectAllRecords();

print\_r($records);

// Delete a record

$deleteResult = $db->deleteRecord(2);

echo $deleteResult . "<br>";

// Close connection

$db->closeConnection();

?>