**Practical 01**

|  |
| --- |
| **Write a PHP program to demonstrate the use of Decision making control structures using-** |
| **a. If statement** |
| **b. If-else statement** |
| **c. Switch statement** |

<?php

// Example of If statement

$number = 10;

if ($number > 0) {

echo "The number is positive.<br>";

}

// Example of If-Else statement

$age = 18;

if ($age >= 18) {

echo "You are an adult.<br>";

} else {

echo "You are not an adult.<br>";

}

// Example of Switch statement

$day = "Monday";

switch ($day) {

case "Monday":

echo "Today is Monday.<br>";

break;

case "Tuesday":

echo "Today is Tuesday.<br>";

break;

case "Wednesday":

echo "Today is Wednesday.<br>";

break;

case "Thursday":

echo "Today is Thursday.<br>";

break;

case "Friday":

echo "Today is Friday.<br>";

break;

case "Saturday":

echo "Today is Saturday.<br>";

break;

case "Sunday":

echo "Today is Sunday.<br>";

break;

default:

**Output-**

The number is positive.  
You are an adult.  
Today is Monday.

echo "Invalid day.<br>";

break;

}

?>

**Practical 02**

|  |
| --- |
| Write a PHP program to demonstrate the use of Looping structures  using- |
| a) While statement |
| b) Do-while statement |
| c) For statement |
| d) Foreach statement |

**a) While statement**

**Output:-**

While loop iteration 1  
While loop iteration 2  
While loop iteration 3  
While loop iteration 4  
While loop iteration 5

<?php

$i = 1;

while ($i <= 5) {

echo "While loop iteration $i<br>";

$i++;

}

?>

**b) Do-while statement**

**Output:-**

Do-while loop iteration 1  
Do-while loop iteration 2  
Do-while loop iteration 3  
Do-while loop iteration 4  
Do-while loop iteration 5

<?php

$j = 1;

do {

echo "Do-while loop iteration $j<br>";

$j++;

} while ($j <= 5);

?>

**c) For statement**

**Output:-**

For loop iteration 1  
For loop iteration 2  
For loop iteration 3  
For loop iteration 4  
For loop iteration 5

<?php

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

echo "For loop iteration $k<br>";

}

?>

**d) Foreach statement**

**<?php**

**$array = ["A", "B", "C", "D", "E"];**

**foreach ($array as $index => $value) {**

**echo "Foreach loop iteration $index with value $value<br>";**

**}**

**Output:-**

Foreach loop iteration 0 with value A  
Foreach loop iteration 1 with value B  
Foreach loop iteration 2 with value C  
Foreach loop iteration 3 with value D  
Foreach loop iteration 4 with value E

**?>**

|  |
| --- |
| **Write a PHP program for creating and manipulating-** |
| **a) Indexed array** |
| **b) Associative array** |
| **c) Multidimensional array** |

**Practical 03**

**a) Indexed array**

<?php

// Creating an indexed array

$fruits = array("Apple", "Banana", "Orange", "Grapes");

// Adding an element to the array

**Output:-**

First fruit: Apple  
All fruits:  
Apple  
Banana  
Orange  
Grapes  
Mango

$fruits[] = "Mango";

// Accessing elements

echo "First fruit: " . $fruits[0] . "<br>";

echo "All fruits:<br>";

foreach ($fruits as $fruit) {

echo $fruit . "<br>";

}

?>

**b) Associative array**

<?php

// Creating an associative array

$age = array("John" => 25, "Jane" => 30, "Sam" => 22);

// Adding an element to the array

$age["Tom"] = 27;

// Accessing elements

echo "Jane's age: " . $age["Jane"] . "<br>";

echo "All ages:<br>";

foreach ($age as $name => $age\_value) {

echo $name . " is " . $age\_value . " years old.<br>";

}

?>

**Output:-**

Jane's age: 30  
All ages:  
John is 25 years old.  
Jane is 30 years old.  
Sam is 22 years old.  
Tom is 27 years old.  
John's Math score: 85

**c) Multidimensional array**

<?php

// Creating a multidimensional array

$students = array(

"John" => array("Math" => 85, "Science" => 90),

"Jane" => array("Math" => 78, "Science" => 85),

"Sam" => array("Math" => 82, "Science" => 88)

);

// Adding an element to the array

$students["Tom"] = array("Math" => 80, "Science" => 87);

// Accessing elements

echo "John's Math score: " . $students["John"]["Math"] . "<br>";

echo "All students' scores:<br>";

foreach ($students as $name => $subjects) {

echo $name . "'s scores:<br>";

foreach ($subjects as $subject => $score) {

echo $subject . ": " . $score . "<br>";

}

}

?>

**Output:-**

All students' scores:  
John's scores:  
Math: 85  
Science: 90  
Jane's scores:  
Math: 78  
Science: 85  
Sam's scores:  
Math: 82  
Science: 88  
Tom's scores:  
Math: 80  
Science: 87

**Practical 04**

|  |
| --- |
| **A. Write a PHP program to-** |
| * **Calculate length of string.** |
| * **Count the number of words in string without using string functions.** |
| **B. Write a simple PHP program to demonstrate use of various built-in string functions.** |

|  |
| --- |
| A. Write a PHP program to- |
| * Calculate length of string. |
| * Count the number of words in string without using string functions. |

<?php

// Calculate length of string

function calculateStringLength($str) {

$length = 0;

while (isset($str[$length])) {

$length++;

}

return $length;

}

$string = "Hello World";

echo "Length of the string: " . calculateStringLength($string) . "<br>";

// Count the number of words in string without using string functions

function countWords($str) {

$words = 0;

$inWord = false;

for ($i = 0; $i < calculateStringLength($str); $i++) {

if ($str[$i] != ' ' && !$inWord) {

$words++;

$inWord = true;

} elseif ($str[$i] == ' ') {

$inWord = false;

}

}

return $words;

}

echo "Number of words in the string: " . countWords($string);

?>

**Output:-**

Length of the string: 11  
Number of words in the string: 2

**B. Write a simple PHP program to demonstrate use of various built-in string functions.**

<?php

$string = "Hello World! Welcome to PHP string functions.";

// Calculate length of string

echo "Length of the string: " . strlen($string) . "<br>";

// Count the number of words in the string

echo "Number of words in the string: " . str\_word\_count($string) . "<br>";

// Convert the string to uppercase

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

// Convert the string to lowercase

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

// Reverse the string

echo "Reversed string: " . strrev($string) . "<br>";

// Find the position of a word in the string

$word = "Welcome";

echo "Position of '$word' in the string: " . strpos($string, $word) . "<br>";

?>

**Output:-**

Length of the string: 45  
Number of words in the string: 7  
Uppercase string: HELLO WORLD! WELCOME TO PHP STRING FUNCTIONS.  
Lowercase string: hello world! welcome to php string functions.  
Reversed string: .snoitcnuf gnirts PHP ot emocleW !dlroW olleH  
Position of 'Welcome' in the string: 13

**Practical 05**

|  |
| --- |
| **Write a simple PHP program to demonstrate use of simple function and parameterized function.** |

<?php

// Simple function to print a greeting

function greet() {

echo "Hello, welcome to the world of PHP!<br>";

}

// Call the simple function

greet();

?>

**Output:-**

Hello, welcome to the world of PHP!

|  |
| --- |
| **Write a PHP program to** |
| **a) Inherit members of super class in subclass.** |
| **b) Create constructor to initialize object of class by using object oriented concepts.** |

**Practical 06**

**a) Inherit members of super class in subclass.**

<?php

// Superclass

class Animal {

public $name;

public function \_\_construct($name) {

$this->name = $name;

}

public function sound() {

return "Some generic animal sound";

}

}

// Subclass

class Dog extends Animal {

public $breed;

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

parent::\_\_construct($name); // Call the parent constructor

$this->breed = $breed;

}

public function sound() {

return "Bark";

}

public function display() {

echo "This is a {$this->breed} named {$this->name} and it makes the sound: " . $this->sound();

}

}

// Create an instance of Dog

$myDog = new Dog("Buddy", "Golden Retriever");

$myDog->display();

?>

**Output:-**

This is a Golden Retriever named Buddy and it makes the sound: Bark

|  |
| --- |
| **Write a simple PHP program on Introspection and Serialization.** |

**Practical 07**

**Introspection**

Introspection in PHP allows you to examine classes, interfaces, properties, and methods at runtime. Here's a simple example:

<?php

class MyClass {

public $property1;

private $property2;

public function \_\_construct($prop1, $prop2) {

$this->property1 = $prop1;

$this->property2 = $prop2;

}

public function myMethod() {

return "Hello, World!";

}

}

$obj = new MyClass("value1", "value2");

// Using introspection to get class information

echo "Class name: " . get\_class($obj) . "<br>";

echo "Methods: " . implode(', ', get\_class\_methods($obj)) . "<br>";

$reflect = new ReflectionClass($obj);

echo "Properties: ";

foreach ($reflect->getProperties() as $property) {

echo $property->getName() . " ";

}

echo "<br>";

?>

**Output:-**

Class name: MyClass  
Methods: \_\_construct, myMethod  
Properties: property1 property2

**Serialization.**

<?php

class User {

public $name;

public $email;

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

$this->name = $name;

$this->email = $email;

}

public function display() {

echo "Name: {$this->name}, Email: {$this->email}<br>";

}

}

$user = new User("John Doe", "john@example.com");

// Serialize the object

$serializedUser = serialize($user);

echo "Serialized User: " . $serializedUser . "<br>";

// Unserialize the object

$unserializedUser = unserialize($serializedUser);

$unserializedUser->display();

?>

**Output:-**

Serialized User: O:4:"User":2:{s:4:"name";s:8:"John Doe";s:5:"email";s:16:"john@example.com";}  
Name: John Doe, Email: john@example.com

**Practical 08**

|  |
| --- |
| **Design a web page using following form controls:** |
| **a. Text box, b. Radio button, c. Check box, d. Buttons** |

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Form Controls Example</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

}

.form-group {

margin-bottom: 15px;

}

label {

display: block;

margin-bottom: 5px;

}

</style>

</head>

<body>

<h1>Form Controls Example</h1>

<form action="submit\_form.php" method="POST">

<!-- Text Box -->

<div class="form-group">

<label for="name">Name:</label>

<input type="text" id="name" name="name">

</div>

<!-- Radio Buttons -->

<div class="form-group">

<label>Gender:</label>

<input type="radio" id="male" name="gender" value="male">

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

<input type="radio" id="female" name="gender" value="female">

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

</div>

<!-- Check Boxes -->

<div class="form-group">

<label>Hobbies:</label>

<input type="checkbox" id="reading" name="hobbies" value="reading">

<label for="reading">Reading</label>

<input type="checkbox" id="traveling" name="hobbies" value="traveling">

<label for="traveling">Traveling</label>

<input type="checkbox" id="cooking" name="hobbies" value="cooking">

<label for="cooking">Cooking</label>

</div>

<!-- Buttons -->

<div class="form-group">

<button type="submit">Submit</button>

<button type="reset">Reset</button>

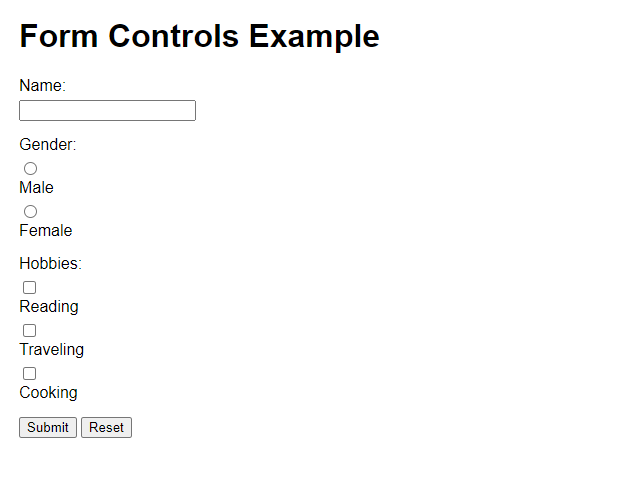
</div>

</form>

</body>

</html>

**Output:-**



|  |
| --- |
| **Design a web page using following form controls:** |
| **a. List box, b. Combo box, c.Hidden field box** |

**Practical 09**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Form Controls Example</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

}

.form-group {

margin-bottom: 15px;

}

label {

display: block;

margin-bottom: 5px;

}

</style>

</head>

<body>

<h1>Form Controls Example</h1>

<form action="submit\_form.php" method="POST">

<!-- List Box -->

<div class="form-group">

<label for="fruits">Select your favorite fruits (Ctrl+Click to select multiple):</label>

<select id="fruits" name="fruits[]" multiple size="4">

<option value="apple">Apple</option>

<option value="banana">Banana</option>

<option value="orange">Orange</option>

<option value="grapes">Grapes</option>

</select>

</div>

<!-- Combo Box (Drop-down Menu) -->

<div class="form-group">

<label for="country">Select your country:</label>

<select id="country" name="country">

<option value="india">India</option>

<option value="usa">USA</option>

<option value="uk">UK</option>

<option value="canada">Canada</option>

</select>

</div>

<!-- Hidden Field -->

<input type="hidden" name="hidden\_value" value="secret\_info">

<!-- Buttons -->

<div class="form-group">

<button type="submit">Submit</button>

<button type="reset">Reset</button>

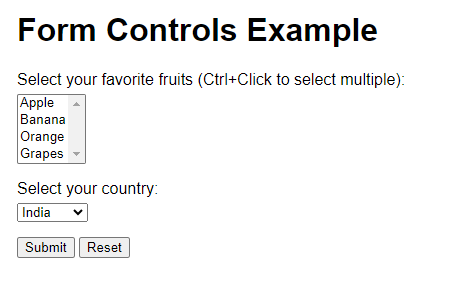
</div>

</form>

</body>

</html>

**Output:-**



**Practical 10**

|  |
| --- |
| **Write simple PHP program to -** |
| **a. Set cookies and read it.** |
| **b. Demonstrate session management** |

**a. Set cookies and read it.**

<?php

// Set a cookie

setcookie("user", "John Doe", time() + (86400 \* 30), "/"); // 86400 = 1 day

echo "Cookie 'user' is set!<br>";

echo "Value is: John Doe<br>";

?>

**Output:-**

Cookie 'user' is set!  
Value is: John Doe

<?php

// Check if the cookie is set

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

echo "Cookie 'user' is set!<br>";

echo "Value is: " . $\_COOKIE["user"] . "<br>";

} else {

echo "Cookie 'user' is not set!";

}

?>

**Output:-**

Cookie 'user' is set!  
Value is: John Doe

<?php

// Start the session

session\_start();

// Set session variables

$\_SESSION["username"] = "JaneDoe";

$\_SESSION["email"] = "jane@example.com";

echo "Session variables are set.<br>";

?>

**Output:-**

Session variables are set.