# Write PHP scripts that demonstrate fundamentals PHP

## Printing "Hello, World!" on the screen:

<?php

echo "Hello, World!";

?>

# Output:-

Hello, World!

## Defining and using variables:

<?php

$name = "Namrata";

$roll = 80;

echo "My name is " . $name . " and My Roll Number is " . $roll;

?>

# Output:-

My name is Namrata and My Roll Number is 80.

## Performing arithmetic operations:

<?php

$num1 = 10;

$num2 = 5;

echo "Addition: " . ($num1 + $num2) . "<br>"; echo "Subtraction: " . ($num1 - $num2) . "<br>"; echo "Multiplication: " . ($num1 \* $num2) . "<br>"; echo "Division: " . ($num1 / $num2) . "<br>";

?>

# Output:-

Addition: 15

Subtraction: 5

Multiplication: 50

Division: 2

## Using conditional statements:

<?php

$num = 10;

if ($num > 0) {

echo "The number is positive.";

} else if ($num < 0) {

echo "The number is negative.";

} else {

echo "The number is zero.";

}

?>

# Output:-

The number is positive.

## Using loops:

<?php

// while loop

$num = 1;

while ($num <= 5) { echo $num . "<br>";

$num++;

}

// for loop

for ($i = 1; $i <= 5; $i++) { echo $i . "<br>";

}

// foreach loop

$colors = array("red", "green", "blue"); foreach ($colors as $color) {

echo $color . "<br>";

}

?>

# Output:-

1

2

3

4

5

1

2

3

4

5

red green blue

## Defining and calling functions:

<?php

function square($num) { return $num \* $num;

}

$result = square(5);

echo "The square of 5 is " . $result;

?>

# Output:-

The square of 5 is 25.

# Write PHP script that will display grade based on criteria given below using the marks obtained in Examination.

<?php

$marks = 85; // replace with the actual marks obtained

if ($marks >= 90) { echo "Grade A+";

} elseif ($marks >= 80) { echo "Grade A";

} elseif ($marks >= 70) { echo "Grade B+";

} elseif ($marks >= 60) { echo "Grade B";

} elseif ($marks >= 50) { echo "Grade C+";

} elseif ($marks >= 40) { echo "Grade C";

} else {

echo "Fail";

}

?>

# Output:-

Grade A.

# Write a PHP script to demonstrate different String functions.

<?php

$string = "The quick brown fox jumps over the lazy dog.";

// Length of the string

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

// Convert string to uppercase

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

// Convert string to lowercase

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

// Replace a substring

echo "Replace 'fox' with 'cat': " . str\_replace("fox", "cat", $string) . "<br>";

// Substring

echo "Substring from index 4 to 15: " . substr($string, 4, 11) . "<br>";

// Split a string into an array

echo "Split string into an array: "; print\_r(explode(" ", $string));

// Join an array into a string

$array = array("The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog."); echo "<br>Join array into a string: " . implode(" ", $array);

?>

# Output:-

Length of the string: 44

Uppercase: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.

Lowercase: the quick brown fox jumps over the lazy dog.

Replace 'fox' with 'cat': The quick brown cat jumps over the lazy dog. Substring from index 4 to 15: quick brown

Split string into an array: Array ( [0] => The [1] => quick [2] => brown [3] => fox [4] => jumps [5]

=> over [6] => the [7] => lazy [8] => dog. )

Join array into a string: The quick brown fox jumps over the lazy dog.

# Write a PHP script to Demonstrate OOPS Concept in PHP.

<?php

// Define a class named 'Person' class Person

{

// Define the properties of the class public $name;

public $age;

// Define a constructor method for the class public function construct($name, $age)

{

$this->name = $name;

$this->age = $age;

}

// Define a method to display the person's name and age public function displayInfo()

{

echo "Name: " . $this->name . "<br>"; echo "Age: " . $this->age . "<br>";

}

}

// Define a class named 'Student' that extends the 'Person' class class Student extends Person

{

// Define additional properties of the class public $rollNo;

public $marks;

// Define a constructor method for the class

public function construct($name, $age, $rollNo, $marks)

{

parent:: construct($name, $age);

$this->rollNo = $rollNo;

$this->marks = $marks;

}

// Define a method to display the student's information public function displayStudentInfo()

{

echo "Name: " . $this->name . "<br>"; echo "Age: " . $this->age . "<br>";

echo "Roll Number: " . $this->rollNo . "<br>"; echo "Marks: " . $this->marks . "<br>";

}

}

// Create an instance of the 'Person' class

$person = new Person("Namrata Patil", 21);

// Call the 'displayInfo()' method of the 'Person' class

$person->displayInfo();

// Create an instance of the 'Student' class

$student = new Student("Lokesh Rajput", 20, "99", 85);

// Call the 'displayInfo()' method of the 'Person' class from the 'Student' class

$student->displayInfo();

// Call the 'displayStudentInfo()' method of the 'Student' class

$student->displayStudentInfo();

?>

# Output:-

Name: Namrata Patil Age: 21

Name: Lokesh Rajput Age: 20

Name: Lokesh Rajput Age: 20

Roll Number: 99

Marks: 85

# Write a PHP script to demonstrate Form Data Handling using Get and Post methods.

1. **POST Method.**

<html>

<body>

<form action="" method="post">

<table border=0>

<tr>

<td>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Student Name | | | | | |
| </td> | | | | | |
| <td> | | | |  | |
| <input type=text name="t1"> | | | | | |
| </td> | | | | | |
| </tr> | | | |  | |
| <tr> | |  | |
| <td> | | | |
| Marks for PHP | | | | | |
| </td> | | | | | |
| <td> | | | |  | |
| <input type=text name="t2"> | | | | | |
| </td> | | | | |  |
| </tr> | | | | |  |
| <tr> | |  | | |
| <td> | | | | |
| Marks for Android | | | | | |
| </td> | | | | | |
| <td> | | | |  | |
| <input type=text name="t3"> | | | | | |
| </td> | | | | | |
| </tr> | | | |  | |
| <tr> | |  | |
| <td> | | | |
| Marks for Cloud Computing | | | | | |
| </td> | | | | | |
| <td> | | | |  | |
| <input type=text name="t4"> | | | | | |
| </td> | | | | |  |
| </tr> | | |  | |
| </table> | | | | |
| <br> |  | | | |
| <br> |

<input type=submit name="s" value="Result">

<?php

if (isset($\_POST['s'])) ////checking whether the input element is set or not

{

$a = $\_POST['t1']; //accessing value from 1st text box

$a1 = $\_POST['t2']; //accessing value from 2nd text field

|  |  |  |
| --- | --- | --- |
| $a2 = $\_POST['t3']; //accessing value from 3rd text field | | |
| $a3 = $\_POST['t4']; //accessing value from 4th text field | | |
| $sum = $a1 + $a2 + $a3; //total marks | |  |
| $avg = $sum / 3; |  | |

if ($avg >= 0 && $avg <= 50)

$grade = "Fail";

if ($avg > 50 && $avg <= 70)

$grade = "C";

if ($avg > 70 && $avg <= 80)

$grade = "B";

if ($avg > 80 && $avg <= 90)

$grade = "A";

if ($avg > 90)

$grade = "E";

echo "<br>";

echo "<font size=4><center>Student is:-" . $a . "</center><br>";

echo "<font size=4><center>Total marks:-" . $sum . "</center><br>";

echo "<font size=4><center>Grade is:-" . $grade . "</center>";

}

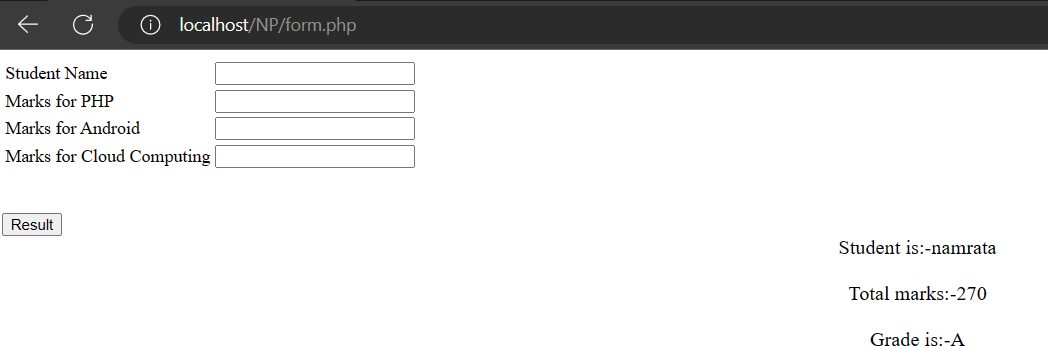
?>

</form>

</body>

</html>

# Output:-



1. **GET Method.**

<html>

<body>

<form action="" method="get">

<table border=0>

<tr>

<td>

Student Name

</td>

<td>

<input type=text name="t1">

</td>

</tr>

<tr>

<td>

Marks for PHP

</td>

<td>

<input type=text name="t2">

</td>

</tr>

<tr>

<td>

Marks for Android

</td>

<td>

<input type=text name="t3">

</td>

</tr>

<tr>

<td>

Marks for Cloud Computing

</td>

<td>

<input type=text name="t4">

</td>

</tr>

</table>

<br>

<br>

<input type=submit name="s" value="Result">

<?php

if (isset($\_GET['s'])) ////checking whether the input element is set or not

{

$a = $\_GET['t1']; //accessing value from 1st text box

$a1 = $\_GET['t2']; //accessing value from 2nd text field

$a2 = $\_GET['t3']; //accessing value from 3rd text field

$a3 = $\_GET['t4']; //accessing value from 4th text field

$sum = $a1 + $a2 + $a3; //total marks

$avg = $sum / 3;

if ($avg >= 0 && $avg <= 50)

$grade = "Fail";

if ($avg > 50 && $avg <= 70)

$grade = "C";

if ($avg > 70 && $avg <= 80)

$grade = "B";

if ($avg > 80 && $avg <= 90)

$grade = "A"; if ($avg > 90)

$grade = "E"; echo "<br>";

echo "<font size=4><center>Student is:-" . $a . "</center><br>"; echo "<font size=4><center>Total marks:-" . $sum . "</center><br>"; echo "<font size=4><center>Grade is:-" . $grade . "</center>";

}

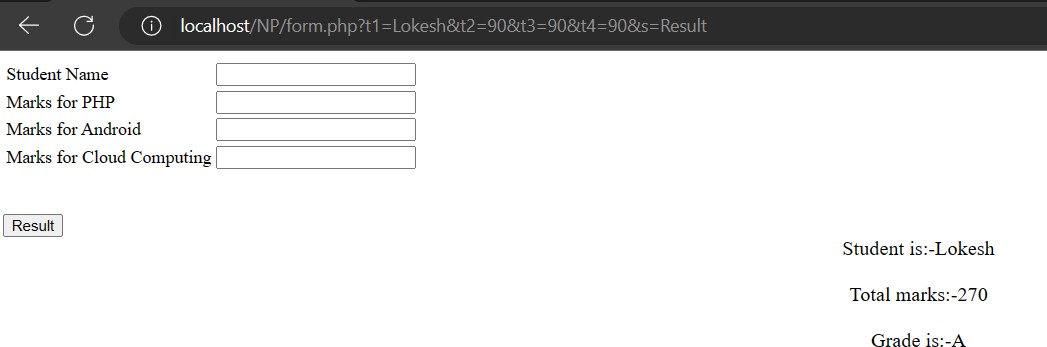
?>

</form>

</body>

</html>

# Output:-



1. **Design a database in MYSQL. Create table in database. Store, Update, Delete and Retrieve data from the table. Display the data from the table.**

## Create a database

CREATE DATABASE library;

## Create tables in the database

CREATE TABLE books (

book\_id INT(11) NOT NULL AUTO\_INCREMENT,

book\_title VARCHAR(255) NOT NULL, author VARCHAR(255) NOT NULL,

publisher VARCHAR(255) NOT NULL, category VARCHAR(255) NOT NULL,

PRIMARY KEY (book\_id)

);

CREATE TABLE users (

user\_id INT(11) NOT NULL AUTO\_INCREMENT,

first\_name VARCHAR(255) NOT NULL, last\_name VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL,

phone\_number VARCHAR(20) NOT NULL, address VARCHAR(255) NOT NULL,

PRIMARY KEY (user\_id)

);

CREATE TABLE borrowed\_books (

borrow\_id INT(11) NOT NULL AUTO\_INCREMENT,

user\_id INT(11) NOT NULL, book\_id INT(11) NOT NULL, borrow\_date DATE NOT NULL, return\_date DATE NOT NULL, PRIMARY KEY (borrow\_id),

FOREIGN KEY (user\_id) REFERENCES users(user\_id), FOREIGN KEY (book\_id) REFERENCES books(book\_id)

);

## Define the fields in the tables

* + The books table has the following fields:

book\_id: an auto-incremented integer that serves as the primary key book\_title: the title of the book

author: the name of the book's author publisher: the name of the book's publisher

category: the category of the book (e.g. science fiction, romance, etc.)

* The users table has the following fields:

user\_id: an auto-incremented integer that serves as the primary key first\_name: the user's first name

last\_name: the user's last name email: the user's email address

phone\_number: the user's phone number address: the user's address

* The borrowed\_books table has the following fields:

borrow\_id: an auto-incremented integer that serves as the primary key user\_id: the ID of the user who borrowed the book

book\_id: the ID of the borrowed book borrow\_date: the date the book was borrowed return\_date: the date the book is due to be returned

## Establish relationships between tables if necessary

**Insert data into the books table:**

INSERT INTO books (book\_title, author, publisher, category)

VALUES ('The Great Gatsby', 'F. Scott Fitzgerald', 'Charles Scribner\'s Sons', 'Classics');

Update data in the **users** table:

UPDATE users

SET phone\_number = '123-456-7890' WHERE user\_id = 1;

## Delete data from the borrowed\_books table:

DELETE FROM borrowed\_books WHERE book\_id = 1**;**

## Retrieve data from the books table:

SELECT \*

FROM books;

# Write a PHP script to store, retrieve and delete cookies on your local machine.

<?php

// Set a cookie

setcookie("username", "John Doe", time() + (86400 \* 30), "/");

// Retrieve a cookie if(isset($\_COOKIE["username"])) {

echo "Welcome " . $\_COOKIE["username"] . "!<br>";

} else {

echo "No cookie found.<br>";

}

?>

# Output:-

Welcome Namrata!

<?php

// Delete a cookie

setcookie("username", "", time() - 3600, "/");

?>

# Output:-

No cookie found.

## Write a PHP script to store, retrieve and delete data using session variables.

<?php session\_start();

// Set session variables

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

$\_SESSION["email"] = "[johndoe@example.com](mailto:johndoe@example.com)";

// Retrieve session variables

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

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

echo "Username: " . $username . "<br>"; echo "Email: " . $email . "<br>";

// Delete session variables unset($\_SESSION["username"]); unset($\_SESSION["email"]);

?>

# Output:-

Username: Lokesh

Email: [lokesh123@gmail.com](mailto:lokesh123@gmail.com)

# Write PHP Script Demonstrate Consrtuctor And Destructor in PHP.

* 1. **Consrtuctor**

<?php class c1

{

public $name; public $id;

public function construct($name, $id)

{

echo $this->name = $name . "<br>"; echo $this->id = $id;

}

}

$c1 = new c1("harshali", 107);

?>

# Output:-

harshali 107

# Destructor

<?php class abc

{

public function hello()

{

echo "hello everyone\n";

}

public function destruct()

{

echo "this is destruct function\n";

}

}

$obj = new abc();

$obj->hello();

?>

# Output:-

hello everyone this is destruct function.

# Write PHP Script Demonstrate Database Connectivity And Insert Data In Database.

* 1. **Create Form register.php**

<html>

<body>

<center>

<form action="register\_a.php" method="post"

<fieldset>

<legend> Sign Up</legend> Name:-

<input type="text" name="name"><br><br> Password

<input type="password" name="pass"><br><br>

<input type="submit" value="Sing Up" name="submit"><br><br>

</fieldset>

</form>

</center>

</body>

</html>

# Connection To Database config.php

<?php

$dbhost = 'localhost';

$dbname = 'bca';

$dbuser = 'root';

$dbpass = '';

$mysqli = mysqli\_connect($dbhost,$dbuser,$dbpass,$dbname);

?>

# Insert Data In Database register\_a.php

<?php include("config.php"); if(isset($\_POST['submit']))

{

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

$password = $\_POST['pass'];

$result = mysqli\_query($mysqli,"insert into abc values('$name','$password')"); if($result)

{

echo "Successfully";

}

else

{

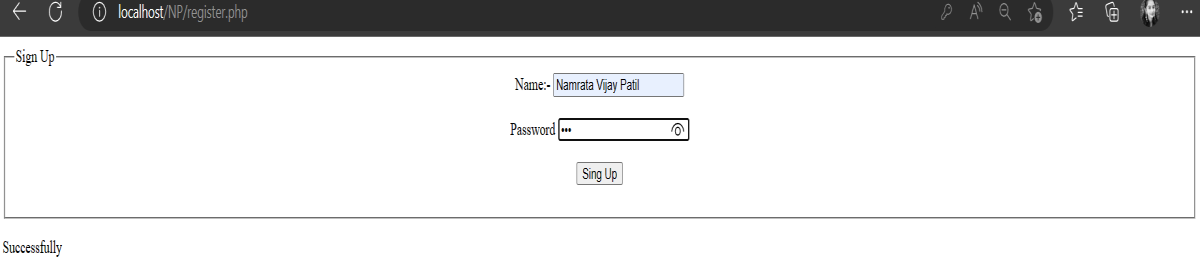
echo "failed";

}

}

?>

# Output:-



1. **Write PHP Script Demonstrate Class And Object.**

<?php

class Myclass

{

public $font\_size ="18px"; public $font\_color = "blue";

public $string\_name = "w3resource"; public function customize\_print()

{

echo "<p style=font-size:".$this->font\_size.";color:".$this->font\_color.";>".$this-

>string\_name."</p>";

}

}

$f = new MyClass;

$f->font\_size = "20px";

$f->font\_color = "red";

$f->string\_name = "Object Oriented Programming"; echo $f->customize\_print();

?>

# Output:-

Object Oriented Programming.