| Course Code: SE-3003 | Course : Web Engineering Lab |
|---|---|

**Spring 2023, Lab Manual – 08**

**LLO 02: Web application using PHP and MySQL**

# Contents:



# Introduction to Web Engineering

Web Engineering is the application of systematic and quantifiable approaches (concepts methods, techniques tools) to cost - effective requirements analysis, design, implementation, testing, operation, and maintenance of **high quality Web applications.**

# Technologies to be studied

- HTML
- CSS
- JavaScript
- Bootstrap
- JQuery
- PHP
- MySQL [Database]
- Laravel [PHP FRAMEWORK]

# Tools – IDEs

- Visual Studio Code
- Adobe Dreamweaver
- Visual Studio

## 7.1 Introduction to PHP

PHP is a general-purpose scripting language geared toward web development. PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. The PHP reference implementation is now produced by The PHP Group. PHP was originally an abbreviation of Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on a variety of operating systems and platforms.

## Creating (Declaring) PHP Variables

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

After the execution of the statements above, the variable $txt will hold the value Hello world!, the variable $x will hold the value 5, and the variable $y will hold the value 10.5. Note: When you assign a text value to a variable, put quotes around the value. Note: Unlike other programming languages, PHP has no command for declaring a variable. It is created the moment you first assign a value to it.

### PHP Variables

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

Rules for PHP variables:

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _ )
- Variable names are case-sensitive ($age and $AGE are two different variables)

- Remember that PHP variable names are case-sensitive!

## PHP is a Loosely Typed Language

In the example above, notice that we did not have to tell PHP which data type the variable is. PHP automatically associates a data type to the variable, depending on its value. Since the data types are not set in a strict sense, you can do things like adding a string to an integer without causing an error In PHP 7, type declarations were added. This gives an option to specify the data type expected when declaring a function, and by enabling the strict requirement, it will throw a "Fatal Error" on a type mismatch.

## PHP echo and print Statements

With PHP, there are two basic ways to get output: echo and print. In this tutorial we use echo or print in almost every example. So, this chapter contains a little more info about those two output statements.

**echo and print** are more or less the same. They are both used to output data to the screen.The differences are small: echo has no return value while print has a return value of 1 so it can be used in expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument. echo is marginally faster than print.

| | "echo" | "print" |
|---|---|---|
| type | This is a type of output string. | This is a type of output string. |
| parenthesis | Not written with parenthesis. | It can or can not be written with parenthesis. |
| strings | It can output one or more strings. | It can output only one string at a time, and returns 1. |
| Functionality | echo is faster than print. | print is slower than echo. |
| argrument | echo($arg1[, $arg2....]) | print($arg) |

## PHP Data Types

Variables can store data of different types, and different data types can do different things.

- PHP supports the following data types:
- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

```
$x = 5985;
var_dump($x);
$x = 10.365;
var_dump($x);
$x = true;
$y = false;
$cars = array("Volvo","BMW","Toyota");
var_dump($cars);
$x = "Hello world!";
$x = null;
var_dump($x);
```

## PHP Object

Classes and objects are the two main aspects of object-oriented programming. A class is a template for objects, and an object is an instance of a class. When the individual objects are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.

Let's assume we have a class named Car. A Car can have properties like model, color, etc. We can define variables like $model, $color, and so on, to hold the values of these properties.

When the individual objects (Volvo, BMW, Toyota, etc.) are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.If you create a __construct() function, PHP will automatically call this function when you create an object from a class.

```
<?php
class Car {
  public $color;
```

```
  public $model;
  public function __construct($color, $model) {
    $this->color = $color;
    $this->model = $model;
  }
  public function message() {
    return "My car is a " . $this->color . " " . $this->model . "!";
  }
}

$myCar = new Car("black", "Volvo");
echo $myCar -> message();
echo "<br>";
$myCar = new Car("red", "Toyota");
echo $myCar -> message();
?>
```

## 7.2 PHP Operators

Operators are used to perform operations on variables and values. PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators
- Conditional assignment operators

| Operator | Name | Example | Result |
|---|---|---|---|
| + | Addition | $x + $y | Sum of $x and $y |
| - | Subtraction | $x - $y | Difference of $x and $y |
| * | Multiplication | $x * $y | Product of $x and $y |
| / | Division | $x / $y | Quotient of $x and $y |
| % | Modulus | $x % $y | Remainder of $x divided by $y |
| ** | Exponentiation | $x ** $y | Result of raising $x to the $y'th power |

| Assignment | Same as... | Description |
|---|---|---|
| x = y | x = y | The left operand gets set to the value of the expression on the right |
| x += y | x = x + y | Addition |
| x -= y | x = x - y | Subtraction |
| x *= y | x = x * y | Multiplication |
| x /= y | x = x / y | Division |
| x %= y | x = x % y | Modulus |

| Operator | Name | Example | Result |
|---|---|---|---|
| == | Equal | $x == $y | Returns true if $x is equal to $y |
| === | Identical | $x === $y | Returns true if $x is equal to $y, and they are of the same type |
| != | Not equal | $x != $y | Returns true if $x is not equal to $y |
| <> | Not equal | $x <> $y | Returns true if $x is not equal to $y |
| !== | Not identical | $x !== $y | Returns true if $x is not equal to $y, or they are not of the same type |
| > | Greater than | $x > $y | Returns true if $x is greater than $y |
| < | Less than | $x < $y | Returns true if $x is less than $y |
| >= | Greater than or equal to | $x >= $y | Returns true if $x is greater than or equal to $y |
| <= | Less than or equal to | $x <= $y | Returns true if $x is less than or equal to $y |

| Operator | Name | Description |
|---|---|---|
| ++$x | Pre-increment | Increments $x by one, then returns $x |
| $x++ | Post-increment | Returns $x, then increments $x by one |
| --$x | Pre-decrement | Decrements $x by one, then returns $x |
| $x-- | Post-decrement | Returns $x, then decrements $x by one |

| Operator | Name | Example | Result |
|---|---|---|---|
| and | And | $x and $y | True if both $x and $y are true |
| or | Or | $x or $y | True if either $x or $y is true |
| xor | Xor | $x xor $y | True if either $x or $y is true, but not both |
| && | And | $x && $y | True if both $x and $y are true |
| \|\| | Or | $x \|\| $y | True if either $x or $y is true |
| ! | Not | !$x | True if $x is not true |

| Operator | Name | Example | Result |
|---|---|---|---|
| . | Concatenation | $txt1 . $txt2 | Concatenation of $txt1 and $txt2 |
| .= | Concatenation assignment | $txt1 .= $txt2 | Appends $txt2 to $txt1 |

| Operator | Name | Example | Result |
|---|---|---|---|
| + | Union | $x + $y | Union of $x and $y |
| == | Equality | $x == $y | Returns true if $x and $y have the same key/value pairs |
| === | Identity | $x === $y | Returns true if $x and $y have the same key/value pairs in the same order and of the same types |
| != | Inequality | $x != $y | Returns true if $x is not equal to $y |
| <> | Inequality | $x <> $y | Returns true if $x is not equal to $y |
| !== | Non-identity | $x !== $y | Returns true if $x is not identical to $y |

| Operator | Name | Example | Result |
|---|---|---|---|
| ?: | Ternary | $x = *expr1* ? *expr2* : *expr3* | Returns the value of $x.<br>The value of $x is *expr2* if *expr1* = TRUE.<br>The value of $x is *expr3* if *expr1* = FALSE |
| ?? | Null coalescing | $x = *expr1* ?? *expr2* | Returns the value of $x.<br>The value of $x is *expr1* if *expr1* exists, and is not NULL.<br>If *expr1* does not exist, or is NULL, the value of $x is *expr2*.<br>Introduced in PHP 7 |

# 7.3 PHP Conditional Statements

Very often when you write code, you want to perform different actions for different conditions. You can use conditional statements in your code to do this. In PHP we have the following conditional statements: if statement - executes some code if one condition is true if...else statement - executes some code if a condition is true and another code if that condition is false if...elseif...else statement - executes different codes for more than two conditions switch statement - selects one of many blocks of code to be executed.

```php
<?php
$t = date("H");

if ($t < "20") {
  echo "Have a good day!";
}
?>
```

```php
<?php
$t = date("H");

if ($t < "10") {
  echo "Have a good morning!";
} elseif ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>
```

## The PHP switch Statement

The switch statement is used to perform different actions based on different conditions. Use the switch statement to **select one of many blocks of code to be executed**.

```php
<?php
$favcolor = "red";

switch ($favcolor) {
  case "red":
    echo "Your favorite color is red!";
    break;
  case "blue":
```

```
    echo "Your favorite color is blue!";
    break;
  case "green":
    echo "Your favorite color is green!";
    break;
  default:
    echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

# 7.4 PHP Loops

Often when you write code, you want the same block of code to run over and over again a certain number of times. So, instead of adding several almost equal code-lines in a script, we can use loops. Loops are used to execute the same block of code again and again, as long as a certain condition is true.

In PHP, we have the following loop types:

- while - loops through a block of code as long as the specified condition is true
- do...while - loops through a block of code once, and then repeats the loop as long as the specified condition is true
- for - loops through a block of code a specified number of times
- foreach - loops through a block of code for each element in an array

```
<?php
$x = 1;

while($x <= 5)

 {
  echo "The number is: $x <br>";
  $x++;
}
?>
```

*Do-while Loop*

```
<?php
$x = 1;

do {
  echo "The number is: $x <br>";
  $x++;
} while ($x <= 5);
?>
```

*For Loop*

```php
<?php
for ($x = 0; $x <= 10; $x++) {
  echo "The number is: $x <br>";
}
?>
```

*Foreach Loop*

```php
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
  echo "$value <br>";
}
?>
```

*Break-Continue in Loop*

```php
<?php
for ($x = 0; $x < 10; $x++) {
  if ($x == 4) {
    continue; //Or Break
  }
  echo "The number is: $x <br>";
}
?>
```

***Function:***

The real power of PHP comes from its functions. PHP has more than 1000 built-in functions, and in addition you can create your own custom functions. Besides the built-in PHP functions, it is possible to create your own functions.

- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute automatically when a page loads.
- A function will be executed by a call to the function.

```php
<?php
function writeMsg() {
  echo "Hello world!";
}

writeMsg(); // call the function
?>
```

## ***PHP Function Arguments***

Information can be passed to functions through arguments. An argument is just like a variable.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma. The following example has a function with one argument ($fname). When the familyName() function is called, we also pass along a name (e.g. Jani), and the name is used inside the function, which outputs several different first names, but an equal last name:

```
<?php
function familyName($fname) {
  echo "$fname Refsnes.<br>";
}

familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("Kai Jim");
familyName("Borge");
?>
```

## ***What is an Array?***

An array is a special variable, which can hold more than one value at a time. An array stores multiple values in one single variable:

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

In PHP, there are three types of arrays:

- **Indexed arrays** - Arrays with a numeric index.
- **Associative arrays** - Arrays with named keys.
- **Multidimensional arrays** - Arrays containing one or more arrays.

# *Task*

**Task 01:** Write a program in PHP that will show the usage of all types of variables [data types] using Loops which discussed in this lab.

**Task 02:** Write a function in PHP that will show the working of operators with respect to the types individually. [Discussed in the lab manual].

**Task 03:** Write a program which will find the factorial of a given number. Exit the program if the input number is negative.

***Example of Factorial: Input number = 5*** ***Factorial is=5*4*3*2*1***

Note: Justify your choice of loop answering two important points: why your choice is optimal? why other looping structure would not be suitable?

**Task 04:** Write a program which will generate the Fibonacci series up to 1000. Also find the sum of the generated Fibonacci numbers divisible by 3, 5 and 7 only.

***Example of Fibonacci series is: 1 1 2 3 5 8 13 25.....***

Note: Do this task by using for loop and while loop. Also identify which one is more efficient?

**Task 05:** Consider an array given below:

A [10] = {5, 4, 3, 2, 1, 6, 10, 9, 7, 8}

Your task is to write a recursive function that searches a given element from that array.

**Task 06:** Write a PHP program to find all roots of a Quadratic equation using switch case AND if else statement also write which one is sufficient and where we prefer whom.

**Task 07:** Create a form[Signup] in PHP to show up the working of global variable GET, POST.

**Task 08:** Create a PHP file and perform all the operations on it which disused in this lab manual.