|  |  |
| --- | --- |
| Name of Student |  |
| Roll No. |  |

**Experiment No: 01**

1. Install and configure PHP, web server, MYSQL
2. Write a program to print “welcome to PHP”
3. Write a program using expression and operators.

**Resources required:**

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| Computer System | Any database tools such as XAMPP |

**Practical Significance:**

PHP is an acronym for "PHP: Hypertext Preprocessor". PHP is a widely-used, open source scripting language. PHP scripts are executed on the server.

**Theoretical Background:**

* A PHP script starts with the <?php and ends with the ?> tag.
* The PHP delimiter <?php and ?> in the following example simply tells the PHP engine to treat the enclosed code block as PHP code, rather than simple HTML.
* On servers with shorthand support enabled you can start a scripting block with <? and end with ?>.

**Syntax:**

<?php

   echo ‘Hello world’;

?>

**Program Code: Write a program to print “welcome to PHP”**

<html>

<body>

<?php echo "Welcome to PHP";

?>

</body>

</html>

Each code line in PHP must end with a semicolon. The semicolon is a separator and is used to distinguish one set of instructions from another.

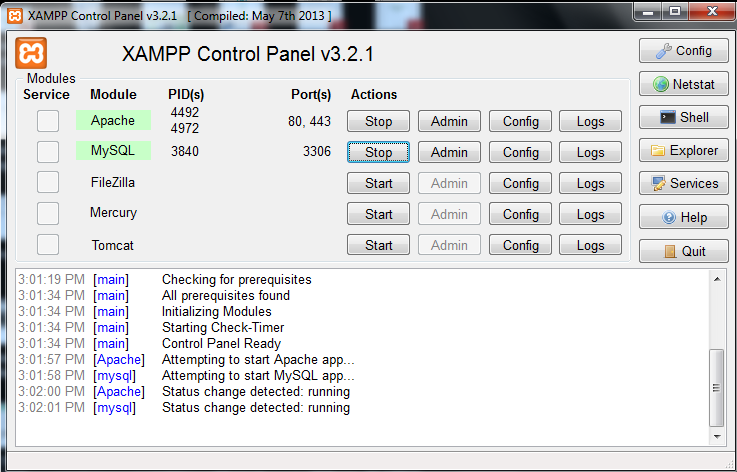
There are two basic statements to output text with PHP: **echo** and **print**. In the example above we have used the echo statement to output the text “Welcome to PHP ".

**How to Run a PHP File in XAMPP**

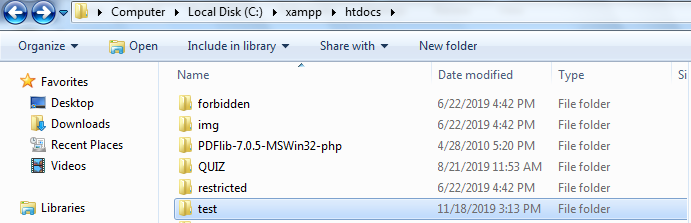
The XAMPP **(Cross**-platform, Apache, MariaDB (Mysql), PHP and Perl) suite of Web development tools, created by Apache Friends, makes it easy to run PHP (Personal Home Pages) scripts locally on your computer. Manual installation of a Web server and PHP requires in-depth configuration knowledge, but installing XAMPP on Windows only requires running an installer package. This package installs not only a Web server and PHP but also MySQL, FileZilla, Mercury, Perl and Tomcat.

**Install XAMPP:**

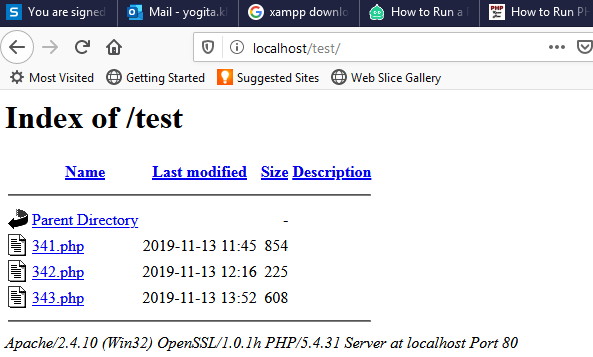
* Go to the Apache Friends website and download XAMPP for Windows. For the easiest install, download the Basic Package's "self-extracting RAR archive." Wait for the download to finish and open it to begin installing XAMPP. Click the "Install" button to start the file extraction. When the Command Prompt screen appears, press the "Enter" key at every question to accept default settings.
* Start the XAMPP program. When started, XAMPP loads itself into your icon tray. The icon is orange with a white bone-like shape in its center. Single-click the icon to expand the Control Panel. Click on the "Start" button next to "Apache" to start your Apache Web server. When Apache is running, the word "Running" will appear next to it, highlighted in green. Also start "MySQL" if your PHP scripts depend on a MySQL database to run.



* Place your PHP files in the "htdocs" folder located under the "XAMMP" folder on your C: drive. The file path is "C:\xampp\htdocs" for your Web server.
* Make sure your PHP files are saved as such; they must have the ".php" file extension. The file path is " C:\xampp\htdocs\test"



* If you create a folder named "test," then use the address "localhost/test" to open them in your browser.



Operators are symbols used to manipulate data stored in variables. A value operated on by an operator is referred to as an operand. The combination of operands with an operator to produce a result is called an expression.

Example: (1+2)

Here integer value 1 and 2 are operands and + is the addition operator, operates on operands to produce the integer result 3.

**Program Code: Write a program using expression and operators.**

<?php

// simple assign operator

$a=20;

echo "a=$a <br/>";

//add then assign operator

$a+=10;

echo "a=a+10 :$a <br/>";

// subtract then assign operator

$a-=10;

echo "a=a-10 :$a <br/>";

// multiply then assign operator

$a\*=10;

echo "a=a\*10 :$a <br/>";

// Divide then assign(quotient) operator

$a/=10;

echo "a=a/10 :$a <br/>";

// Divide then assign(remainder) operator

$a%=2;

echo "a=a%2 :$a <br/>";

?>

**Output:**

a=20   
a=a+10 :30   
a=a-10 :20   
a=a\*10 :200   
a=a/10 :20   
a=a%2 :0

PHP can:

* generate dynamic page content
* create, open, read, write, delete, and close files on the server
* collect form data
* send and receive cookies
* add, delete, modify data in your database
* used to control user-access
* can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

* PHP runs on different platforms (Windows, Linux, Unix, Mac OS X, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.)
* PHP supports a wide range of databases.
* PHP is easy to learn and runs efficiently on the server side.

**Practical related questions:**

1. Sate difference between echo() and print().
2. State the difference between $ and $$ in PHP.
3. State the use of spaceship operator.
4. What are string operators available in PHP?
5. What is the use of var\_dump() in php?
6. Analyze the difference between programming language and scripting language/

**Exercise:** Execute the following script and attach the output:

|  |  |
| --- | --- |
| <?php  $x = -12;  echo ($x > 0) ? “The number is positive”: “The number is negative”;  ?> | <?php  $y = 2;  if (\*\*$y == 4)  {  echo $y;  }  ?> |
| <?php  $x = "test";  $y = "this";  $z = "also";  $x .= $y .= $z ;  echo $x;  echo $y;  ?> | <?php  $a = 10;  echo ++$a;  echo $a++;  echo $a;  echo ++$a;  ?> |

**Experiment No: 02**

**Write a PHP program to demonstrate the use of looping structures using**

1. while statement
2. Do-while else statement
3. for statement
4. for-each statement

**Resources required:**

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| Computer System | Any database tools such as XAMPP |

**Practical Significance:**

Generally instructions are executed sequentially. In some cases it is necessary to change the sequence of executions based on certain conditions. For this purpose decision control structure is required.

**Theoretical Background:**

**a) if statement**

The if statement is used to execute a block of code only if the specified condition evaluates to true.

**Syntax:**

if(condition)

{  
    // Code to be executed  
}

**b) if-else Statement:**

If...else statement first checks the condition. If condition is true, then true statement block is executed. If condition is false, then false statement block is executed.

**Syntax:**

if (condition)

{

// if TRUE then execute this code

}

else

{

// if FALSE then execute this code

}

**c) Nested-if Statement:**

Nested if statements mean an if block inside another if block. Nested if else statement used when we have more than two conditions. It is also called if else if statement.

**Syntax:**

if(condition1)

{  
 // Code to be executed if condition1 is true  
}

elseif(condition2)

{  
    // Code to be executed if the

condition1 is false and condition2 is true  
 }

else

{  
// Code to be executed if both condition1 and condition2 are false  
}

1. **Switch Statement**

The switch-case statement is an alternative to the if-elseif-else statement, which does almost the same thing. The switch-case statement tests a variable against a series of values until it finds a match, and then executes the block of code corresponding to that match. The switch statement is used to avoid long blocks of if..elseif..else code.

**Syntax:**

switch(n)

{

case statement1:

//code to be executed if n==statement1;

break;

case statement2:

//code to be executed if n==statement2;

break;

case statement3:

//code to be executed if n==statement3;

break;

case statement4:

//code to be executed if n==statement4;

break;

......

default:

//code to be executed if n != any case;

}

**Program Code: Write a program for whether number is positive or not.**

<?php

$a=-10;

if ($a > 0)

{

echo "The number is positive";

}

else

{

echo "The number is negative";

}

?>

**Output:**

The number is negative

**Program Code: Write a program to demonstrate the use of Switch statement.**

<?php

$x=-1;

switch($x) {

case 1:

echo "This is case No 1.";

break;

case 2:

echo "This is case No 2.";

break;

case 3:

echo "This is case No 3.";

break;

case 4:

echo "This is case No 4.";

break;

default:

echo "This is default.";

}

?>

Output:

This is default.

**Practical related questions:**

1. How we use if..else and elseif statement in PHP?
2. Difference between if…else and switch statement.

**Exercise:**

1. Write a PHP code to perform arithmetic operations using switch case.
2. Difference between if…else and ternary operator.
3. Why break and continue statement used in php?
4. Write a code to find out whether the year is leap year or not.

**Experiment No: 03**

**Write a PHP program to demonstrate the use of looping structures using**

1. while statement
2. Do-while else statement
3. for statement
4. for-each statement

**Resources required:**

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| Computer System | Any database tools such as XAMPP |

**Practical Significance:**

A loop causes a section of a program to be repeated a certain number of times. The repetition continues while the condition set for it remains true. When the condition becomes false, the loop ends and the control is passed to the statement following the loop. Loop in PHP is used to execute a statement or a block of statements, multiple times until and unless a specific condition is met. This helps the user to save both time and effort of writing the same code multiple times.

**Theoretical Background:**

**a) While Statement:**

The while statement will execute a block of code if and as long as a test condition is true. The while is an entry controlled loop statement. i.e., it first checks the condition at the start of the loop and if its true then it enters the loop and executes the block of statements, and goes on executing it as long as the condition holds true.

**Syntax:**

while (if the condition is true)

{

// code is executed

}

**b) do-while Statement**

This is an exit control loop which means that it first enters the loop, executes the statements, and then checks the condition. Therefore, a statement is executed at least once on using the do…while loop. After executing once, the program is executed as long as the condition holds true.

**Syntax:**

do

{

//code is executed

} while (if condition is true);

**c) for Statement:**

The for statement is used when you know how many times you want to execute a statement or a block of statements. That is, the number of iterations is known beforehand. These type of loops are also known as entry-controlled loops. There are three main parameters to the code, namely the initialization, the test condition and the counter.

**Syntax:**

for (initialization expression; test condition; update expression)

{

// code to be executed

}

**d) For-each statement:**

**foreach loop** is used for array and objects. For every counter of loop, an array element is assigned and the next counter is shifted to the next element.

**Syntax:**

foreach (array\_element as value)

{

//code to be executed

}

**Program Code: Write a program in PHP to calculate Square Root of a number.**

<?php

function my\_sqrt($n)

{

$x = $n;

$y = 1;

while($x > $y)

{

$x = ($x + $y)/2;

$y = $n/$x;

}

return $x;

}

print\_r(my\_sqrt(16)."<br/>");

print\_r(my\_sqrt(144)."<br/>");

?>

Output:

4

12

**Program Code:**

**Write a program in PHP to display content of array using for-each loop.**

<?php

$arr = array (10, 20, 30, 40, 50);

foreach ($arr as $i)

{

echo "$i <br/>";

}

?>

**Output:**

10   
20   
30   
40   
50

**Practical related questions:**

1. Write PHP program to print Fibonacci series.
2. Write a PHP program to print prime number up to n.
3. Difference between for and for-each loop.
4. Difference between while and do-while loop.

**Exercise:**

1. Write the output for following script:

|  |  |
| --- | --- |
| <?php  for ($x = 0; $x <= 10; print ++$x)  {  print ++$x;  }  ?> | <?php  $i = 0;  for ($i)  {  print $i;  }  ?> |
| <?php  for ($x = -1; $x < 10;--$x)  {  print $x;  }  ?> | <?php  for ($x = 1; $x < 10;++$x)  {  print "\*\t";  }  ?> |

1. Create a script to construct the following pattern, using nested for loop.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

3. Create a script to construct the following pattern, using a nested for loop.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

4. Write a PHP script that creates the following table using for loops.

Add cellpadding="3px" and cellspacing="0px" to the table tag.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 \* 1 = 1 | 1 \* 2 = 2 | 1 \* 3 = 3 | 1 \* 4 = 4 | 1 \* 5 = 5 |
| 2 \* 1 = 2 | 2 \* 2 = 4 | 2 \* 3 = 6 | 2 \* 4 = 8 | 2 \* 5 = 10 |
| 3 \* 1 = 3 | 3 \* 2 = 6 | 3 \* 3 = 9 | 3 \* 4 = 12 | 3 \* 5 = 15 |
| 4 \* 1 = 4 | 4 \* 2 = 8 | 4 \* 3 = 12 | 4 \* 4 = 16 | 4 \* 5 = 20 |
| 5 \* 1 = 5 | 5 \* 2 = 10 | 5 \* 3 = 15 | 5 \* 4 = 20 | 5 \* 5 = 25 |
| 6 \* 1 = 6 | 6 \* 2 = 12 | 6 \* 3 = 18 | 6 \* 4 = 24 | 6 \* 5 = 30 |
| 7 \* 1 = 7 | 7 \* 2 = 14 | 7 \* 3 = 21 | 7 \* 4 = 28 | 7 \* 5 = 35 |
| 8 \* 1 = 8 | 8 \* 2 = 16 | 8 \* 3 = 24 | 8 \* 4 = 32 | 8 \* 5 = 40 |
| 9 \* 1 = 9 | 9 \* 2 = 18 | 9 \* 3 = 27 | 9 \* 4 = 36 | 9 \* 5 = 45 |
| 10 \* 1 = 10 | 10 \* 2 = 20 | 10 \* 3 = 30 | 10 \* 4 = 40 | 10 \* 5 = 50 |

5. Write a PHP script that creates the following table (use for loops).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
| 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90 |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

**https://www.w3resource.com/php-exercises/php-for-loop-exercises.php#editorr**

**Experiment No: 04**

**Write a PHP program for creating and manipulating:**

1. Indexed array
2. Associative array
3. Multidimensional array

**Resources required:**

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| Computer System | Any database tools such as XAMPP |

**Practical Significance:**

* Arrays in PHP is a type of data structure that allows to store multiple elements of similar data type under a single variable thereby saving us the effort of creating a different variable for every data.
* An [array](https://www.php.net/manual/en/language.types.array.php) in PHP is actually an *ordered map.* A map is a type that associates *values* to *keys*.
* The arrays are helpful to create a list of elements of similar types, which can be accessed using their index or key.

**Theoretical Background:**