Instructor’s Manual

Introduction to PHP through IDE

IBM101

Authors: Nihar Ranjan Roy, Shahil Raheja & Arvendra Kumar
4/5/2013

Version 0.2
1. Introduction to basic HTML tags.
2. Write a program to check and print whether a given number is even or odd.
3. Write a program to compute net amount from the given quantity purchased and rate per quantity.
   Discount @10% is allowed if the quantity purchased exceeds 100.
4. Write a program to find largest among three numbers using ternary operators.
5. Write a program to print sum of digits of a given number. (using while loop)
6. Write a program to print Fibonacci series up to a given number.
7. Write a program to enter numbers till the user wants. At the end it should display the count of
   positive, negative and zeros entered. (Using do-while loop)
8. Write a function countWords ($str) that takes any string of characters and finds the Number of times
   each word occurs. You should ignore the distinction between capital and lowercase letters.
9. Create a form with one text field and submit buttons for string length, string reverse and uppercase,
   lowercase, string replace. Display the result accordingly.
10. Write a menu-driven program to implement a calculator which performs only addition, subtraction,
    multiplication and division. The operation should happen based on the user choice. (use switch case)
11. Write a function to swap two string values using call by value and call by references.
12. Write a program that will accept an array of integers as input, and output an array where for each
    item in the source array, the new array will perform the following operations:
        - For even numbers divide by 2
        - For odd numbers multiply by 3
13. Create an associative array using the countries as keys, the cities as values and transform it into 2-
    dimensional array and display the data as a table.
14. Given two strings A and B, how would you find out if the characters in B were a subset of the
    characters in A?
15. Write a program that create a file and write contents to it and display it. Then append some data to it.
16. Create a login form with two text fields called “login” and “password”. When user enters “Galgotias”
    as a user name and “university” as a password it should be redirected to a Welcome.HTML page or to
    Sorry.HTML in case of wrong username/password.
17. Write a PHP program using Java Script to convert the decimal number to its binary equivalent. You
    must use a form to accept the number from the user.
18. Write a PHP code that define class Student with attributes RollNo, Name, Branch, and Year, create 3
    instances of it, sets the values of each instance appropriately and print the values of all attributes.
19. Write a function calculateAverage () which takes four int arguments which are marks for four
    courses in the semester and returns their average as a float. The calculateAverage () function should
    take only valid range for marks which is between 0 - 100. If the marks are out of range is should
    throw an OutOfRangeException and handles it.
20. Create a form with a text box asking to enter your favorite city with a submit button when the user
    enters the city and clicks the submit button another php page should be opened displaying “Welcome
    to the city”.

******
Table of Contents

1. Introduction to basic HTML tags. ................................................................. 1
2. Write a Program to check and print whether a given number is even or odd. ........ 3
3. Write a program to compute net amount from the given quantity purchased and rate per quantity. Discount @10% is allowed if the quantity purchased exceeds 100. ........................................ 4
4. Write a program to find largest among three numbers using ternary operator ........ 5
5. Write a program to print sum of digits of a given number. (using while loop) ....... 6
6. Write a program to print Fibonacci series upto a given number. .......................... 8
7. Write a program to enter numbers till the user wants. At the end it should display the count of positive, negative and zeros entered. (Using do-while loop) ..................................... 10
8. Write a function countWords ($str) that takes any string of characters and finds the number of times each word occurs. You should ignore the distinction between capital and lowercase letters. .......... 11
9. Create a form with one text field and submit buttons for string length, string reverse and uppercase, lowercase, string replace. Display the result accordingly............................... 14
10. Write a Menu-Driven program to implement a calculator which performs only addition, subtraction, multiplication and division. The operation should happen based on the user choice. (use switch case) ................................................................. 17
11. Write a function to swap two string values using call by value and call by references .................... 20
12. Write a program that will accept an array of integers as input, and output an array where for each item in the source array, the new array will perform the following operations: (a) For even numbers divide by 2 (b) For odd numbers multiply by 3 ................................................................. 22
13. Create an associative array using the countries as keys, the cities as values and transform it into 2-dimensional array and display the data as a table. ................................................................. 23
14. Given two strings A and B, how would you find out if the characters in B were a subset of the characters in A? ......................................................................................... 25
15. Write a program that creates a file and writes contents to it and display it. Then append some data to it. .................................................................................................................. 26
16. Create a login form with two text fields called “login” and “password”. When user enters “Galgotias” as a user name and “university” as a password it should be redirected to a Welcome.HTML page or to Sorry.HTML in case of wrong username/password.................................................. 28
17. Write a PHP program using Java Script to convert the decimal number to its binary equivalent. You must use a form to accept the number from the user. ................................................................. 30
18. Write a PHP code that define class Student with attributes RollNo, Name, Branch, and Year, create 3 instances of it, sets the values of each instance appropriately and print the values of all attributes. 32
19. Write a function calculateAverage () which takes four int arguments which are marks for four courses in the semester and returns their average as a float. The calculateAverage () function should take only valid range for marks which is between 0 - 100. If the marks are out of range is should throw an OutOfRangeException and handles it........................................ 34
20. Create a form with a text box asking to enter your favorite city with a submit button when the user enters the city and clicks the submits button another php page should be opened displaying “Welcome to the city”. ............................................................. 35
SOLUTIONS TO EXPERIMENTS

EXPERIMENT NO 1:

AIM OF EXPERIMENT: Introduction to basic HTML tags.

OBJECTIVE: Introduction to basic HTML tags. Such as

1. <HTML>
2. <BODY>
3. <BR>
4. <TABLE>
5. <A HREF>
6. <IMG>
7. <B>
8. <I>
9. <P>
10. <SUB>
11. <SUP>

THEORY:
The HTML code for a link is simple. It looks like this:
<a href="url">Link text</a>
The href attribute specifies the destination of a link

Syntax for defining an image:
<img src="url" alt="some_text">

SOURCE CODE:

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

<table>
<thead>
<tr>
<th>Table border=&quot;1&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;tr&gt;</td>
</tr>
<tr>
<td>&lt;th&gt;Header 1&lt;/th&gt;</td>
</tr>
<tr>
<td>&lt;th&gt;Header 2&lt;/th&gt;</td>
</tr>
<tr>
<td>&lt;/tr&gt;</td>
</tr>
<tr>
<td>&lt;tr&gt;</td>
</tr>
<tr>
<td>&lt;td&gt;row 1, cell 1&lt;/td&gt;</td>
</tr>
<tr>
<td>&lt;td&gt;row 1, cell 2&lt;/td&gt;</td>
</tr>
<tr>
<td>&lt;/tr&gt;</td>
</tr>
<tr>
<td>&lt;tr&gt;</td>
</tr>
<tr>
<td>&lt;td&gt;row 2, cell 1&lt;/td&gt;</td>
</tr>
<tr>
<td>&lt;td&gt;row 2, cell 2&lt;/td&gt;</td>
</tr>
<tr>
<td>&lt;/tr&gt;</td>
</tr>
<tr>
<td>&lt;/table</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>My First Heading</th>
</tr>
</thead>
<tbody>
<tr>
<td>My first paragraph.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>How the HTML code looks in your browser:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Header 1</strong></td>
</tr>
<tr>
<td>row 1, cell 1</td>
</tr>
<tr>
<td>row 2, cell 1</td>
</tr>
</tbody>
</table>

```
<html>
<body>
<p><b>This text is bold</b></p>
<p><strong>This text is italic</strong></p>
</body>
</html>
```

<table>
<thead>
<tr>
<th>This text is bold</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is computer output</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>This text is strong</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is computer output</td>
</tr>
</tbody>
</table>
<p><i>This text is italic</i></p>
<p><em>This text is emphasized</em></p>
<p><code>This is computer output</code></p>
<p>This is sub<sub>script</sub> and sup<sup>erscript</sup></p>
**Experiment No 2:**

**AIM OF EXPERIMENT:** Write a Program to check and print whether a given number is even or odd.

**Objective**

Illustrate use of
- modulo operator.
- If statement

**Theory (If Any)**

**Source Code:**

Case 1: When number is given

```html
<html>
<body>
<?php
    // Program to check and print whether a given number is even or odd.
    $n=10;
    if ($n%2==0)
        echo "Given number is even";
    else
        echo "Given number is odd";
?>
</body>
</html>
```

Running the program: http://localhost/evenodd.php

**Output:**

Given number is even

**Case 2: When number is input through the keyboard**

```php
<?php
    // Program to check and print whether an entered number is even or odd.
    echo "Enter number";
    $n=fgets (STDIN);
    if ($n%2==0)
        echo "Entered number is even";
    else
        echo "Entered number is odd";
?>
```

Running the program: C:\XAMPP\PHP>php evenodd.php

**Output**

Enter number 5
Entered number is odd
EXPERIMENT NO 3:

AIM OF EXPERIMENT: Write a program to compute net amount from the given quantity purchased and rate per quantity. Discount @10% is allowed if the quantity purchased exceeds 100.

OBJECTIVE:
illustrate use of
• fgets (STDIN)
• how to compile & run the code at command prompt (non-web)

THEORY (IF ANY)
Syntax: string fgets ( resource $handle [, int $length ] )
Gets a line from file pointer.

A php program named netamount.php can be run from the command prompt like this
C:\XAMPP\PHP>php netamount.php

SOURCE CODE:
```php
<?php
    echo "Enter quantity purchased";
    $quantity=fgets (STDIN);
    echo "Enter rate per item";
    $rate=fgets (STDIN);
    if ($quantity>100)
        $discount=10;
    else
        $discount=0;
    $netamount= ($quantity*$rate)-($quantity*$rate*$discount/100);
    echo "Net payable amount is". $netamount;
?>
```

Running the program:
C:\XAMPP\PHP>php netamount.php

OUTPUT:

• Enter quantity purchased 100
  Enter rate per item 10
  Net payable amount is 1000
• Enter quantity purchased 101
  Enter rate per item 10
  Net payable amount is 909
**Experiment No 4:**

**Aim of Experiment:** Write a program to find largest among three numbers using ternary operator.

**Objective:**
Illustrate use of
- ternary operator

**Theory (if Any):**
For running the program with the url the file must be placed in htdocs folder and the following url can be typed in the web browser.

http://localhost/largest.php

where largest.php is the name of the php file.

**Source Code:**

```
<html>
<body>
<?php
$a=20;
$b=30;
$c=25;
$large= ($a>$b)? (($a>$c)? $a: $c) :(( $b>$c)? $b: $c);
echo "largest number is". $large;
?>
</body>
</html>
```

**Output:**
Largest number is 30
**EXPERIMENT NO 5:**

**AIM OF EXPERIMENT:** Write a program to print sum of digits of a given number. (using while loop)

**OBJECTIVE:**

Illustrate use of

While loop

**THEORY (IF ANY)**

*while* loops are the simplest type of loop in PHP. They behave just like their C counterparts. The basic form of a *while* statement is:

```php
while (expr)
    statement
```

The meaning of a *while* statement is simple. It tells PHP to execute the nested statement(s) repeatedly, as long as the *while* expression evaluates to **TRUE**. The value of the expression is checked each time at the beginning of the loop, so even if this value changes during the execution of the nested statement(s), execution will not stop until the end of the iteration (each time PHP runs the statements in the loop is one iteration). Sometimes, if the *while* expression evaluates to **FALSE** from the very beginning, the nested statement(s) won't even be run once.

Like with the *if* statement, you can group multiple statements within the same *while* loop by surrounding a group of statements with curly braces, or by using the alternate syntax:

```php
while (expr):
    statement
    ...
endwhile;
```

The following examples are identical, and both print the numbers 1 through 10:

```php
<?php
/* example 1 */

$i = 1;
while ($i <= 10) {
    echo $i++; /* the printed value would be $i before the increment (post-increment) */
}

/* example 2 */

$i = 1;
while ($i <= 10):
    echo $i;
    $i++;
endwhile;
?>
```
**Source Code:**

```php
<?php
    echo "Enter number";
    $a=fgets (STDIN);
    $sum=0;
    while ($a>0)
    {
        $r=$a%10;
        $sum=$sum+$r;
        $a=$a/10;
    }
    echo "Sum of the digits is". $sum;
?>
```

**Output:**

Enter number 584
Sum of the digits is 17
Enter number 5
Sum of the digits is 5
EXPERIMENT NO 6:

AIM OF EXPERIMENT: Write a program to print Fibonacci series upto a given number.

OBJECTIVE:
Illustrate the use of for loop, and check the ability to apply the things learnt in last class/lab

THEORY (IF ANY)
In mathematics, the Fibonacci numbers or Fibonacci series or Fibonacci sequence are the numbers in the following integer sequence:

0 1 1 2 3 5 8 13 21 ......
By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two.

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation

\[ F_n = F_{n-1} + F_{n-2}, \]

with seed values[3]

\[ F_0 = 0, \quad F_1 = 1. \]

The Fibonacci sequence is named after Leonardo of Pisa, who was known as Fibonacci.

SOURCE CODE:

```php
<?php
    echo "Enter number of terms you want to print the Fibonacci series";
    $num=fgets (STDIN);
    $a=0;
    $b=1;
    if ($num<0)
        echo "Series not possible";
    else if ($num==1)
        echo $a;
    else if ($num==2)
        echo $a."\n".$b;
    else
    {
        echo $a."\n". $b;
        for ($i=3; $i<=$num; $i=$i+1)
        {
            $c=$a+$b;
            echo "\n". $c;
            $a=$b;
            $b=$c;
        }
    }
?>
```
**OUTPUT:**

Enter number of terms you want to print the Fibonacci series 2
0
1

Enter number of terms you want to print the Fibonacci series 6
0
1
1
2
3
**EXPERIMENT NO 7:**

**AIM OF EXPERIMENT:** Write a program to enter numbers till the user wants. At the end it should display the count of positive, negative and zeros entered. (Using do-while loop)

**OBJECTIVE:**
Illustrate the use of
- Command line arguments
- Count function
- Do while loop

**THEORY (IF ANY)**
count() function counts elements in an array.
$argv is a predefined array that stores the command line arguments.
do-while loops are very similar to while loops, except the truth expression is checked at the end of each iteration instead of in the beginning. The main difference from regular while loops is that the first iteration of a do-while loop is guaranteed to run (the truth expression is only checked at the end of the iteration), whereas it may not necessarily run with a regular while loop (the truth expression is checked at the beginning of each iteration, if it evaluates to FALSE right from the beginning, the loop execution would end immediately).
There is just one syntax for do-while loops:
```php
<?php
$i = 0;
do {
    echo $i;
} while ($i > 0);
?>
```
The above loop would run one time exactly, since after the first iteration, when truth expression is checked, it evaluates to FALSE ($i is not bigger than 0) and the loop execution ends.

**SOURCE CODE:**
```php
<?php
$i= count($argv);
$p=0;$n=0;$z=0;
for($j=1;$j<$i;$j++)
{
    if($argv[$j]>0)
        $p++;
    else if ($argv[$j]==0)
        $z++;
    else
        $n++;    
} echo "Positive nos=".$p;
echo "Negative nos=".$n;
echo "Zeros =".$z;
?>
```

**OUTPUT:**
```
php hello.php 2 4 5 -3 -4 0 2 0 3
Positive nos=5 Negative nos=2 Zeros =2
```
**Experiment No 8:**

**Aim of Experiment:** Write a function `countWords($str)` that takes any string of characters and finds the number of times each word occurs. You should ignore the distinction between capital and lowercase letters.

**Objective:**

Illustrate how to write functions in PHP and use of `explode` function.

**Theory (if any):**

A function will be executed by a call to the function.

Syntax

```
function functionName()
{
  code to be executed;
}
```

PHP function guidelines:

- Give the function a name that reflects what the function does
- The function name can start with a letter or underscore (not a number)

Example

A simple function that writes my name when it is called:

```
<html>
<body>

<?php
function writeName()
{
  echo "Kai Jim Refsnes";
}

echo "My name is ";
writeName();
?>

</body>
</html>
```

Output:
My name is Kai Jim Refsnes

To add more functionality to a function, we can add parameters. A parameter is just like a variable.

Parameters are specified after the function name, inside the parentheses.

Example 1

The following example will write different first names, but equal last name:

```
<html>
<body>

<?php
function writeName($fname)
{

```
Example 1

The following function has one parameter:

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

```php
echo "My name is ";
writeName("Kai Jim");
echo "My sister's name is ";
writeName("Hege");
echo "My brother's name is ";
writeName("Stale");
?>
```

```html
</body>
</html>
```

Output:
My name is Kai Jim Refsnes.
My sister's name is Hege Refsnes.
My brother's name is Stale Refsnes.

Example 2

The following function has two parameters:

```php
function writeName($fname,$punctuation)
{
    echo $fname . " Refsnes" . $punctuation . "<br>
}
```

```php
echo "My name is ";
writeName("Kai Jim",".");
echo "My sister's name is ";
writeName("Hege","!");
echo "My brother's name is ";
writeName("Ståle","?");
?>
```

```html
</body>
</html>
```

Output:
My name is Kai Jim Refsnes.
My sister's name is Hege Refsnes!
My brother's name is Ståle Refsnes?
**Source Code**

```php
<?php
    function countStr($string,$substring)
    {
        $cArr = explode($substring,$string);
        $substring_count = count($cArr) - 1;
        echo $substring_count;
    }

    $string = 'when the men get the hen';
    $substring = 'he';
    Countstr($string,$substring);

?>

**OUTPUT:**

4
EXPERIMENT NO 9:
AIM OF EXPERIMENT: Create a form with one text field and submit buttons for string length, string reverse and uppercase, lowercase, string replace. Display the result accordingly.

OBJECTIVE:
Illustrate the use of
- basic string functions like strlen, strtoupper, strtolower and strrev.
- form tag in html
- use of $_GET variable

THEORY (IF ANY):

int strlen ( string $string )
Returns the length of the given string

string strtoupper ( string $string )
Returns string with all alphabetic characters converted to uppercase.

string strtolower ( string $str )
Returns string with all alphabetic characters converted to lowercase.

string strrev ( string $string )
Returns string, reversed.

Form tag

The <form> tag tells the browser where the form starts and ends. You can add all kinds of HTML tags between the <form> and </form> tags.

Look at this example:

<html>
<head>
<title>My Page</title>
</head>

<body>
<!-- Here goes HTML -->
<form>
<!-- Here goes form fields and HTML -->
</form>
<!-- Here goes HTML -->
</body>
</html>

Note: Unlike a table, forms are not visible on the page.

The address is the url of the php script the content should be sent to. The post and get methods are simply two different methods for submitting data to the script.

If you are using a pre-programmed script (which we assume here) it is not important to understand the difference between get and post.
In the description of the script you are using it will be made clear whether the scripts should be addressed using one method or the other.

Below is an example of a typical form tag, with both action and method specified.

```html
<html>
<head>
<title>My Page</title>
</head>

<body>
<!-- Here goes HTML -->
<form method="post" action="target.php">
<!-- Here goes form fields and HTML -->
</form>
<!-- Here goes HTML -->
</body>
</html>

$_GET variable

The predefined $_GET variable is used to collect values in a form with method="get"

Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send.

Example

```html
<form action="welcome.php" method="get">
Name: <input type="text" name="fname">
Age: <input type="text" name="age">
<input type="submit">
</form>
```

When the user clicks the "Submit" button, the URL sent to the server could look something like this: http://localhost/welcome.php?fname=Peter&age=37

The "welcome.php" file can now use the $_GET variable to collect form data (the names of the form fields will automatically be the keys in the $_GET array):

Welcome <?php echo $_GET["fname"];?>.<br>
You are <?php echo $_GET["age"];?> years old!
**Source Code:**

<table>
<thead>
<tr>
<th>Html file source code</th>
<th>Str.php file source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;html&gt;</td>
<td>&lt;?php</td>
</tr>
<tr>
<td>&lt;body&gt;</td>
<td>$str=$_GET[&quot;t1&quot;];</td>
</tr>
<tr>
<td>&lt;form method=get action=&quot;str.php&quot;&gt;</td>
<td>echo &quot;string length is&quot; . strlen($str); echo &quot;&lt;br&gt;&quot;;</td>
</tr>
<tr>
<td>Please enter the string &lt;br&gt; &lt;input type=text name=&quot;t1&quot;&gt;&lt;br&gt;</td>
<td>echo &quot;string upper&quot; . strtoupper($str);echo &quot;&lt;br&gt;&quot;;</td>
</tr>
<tr>
<td>&lt;input type=submit&gt;</td>
<td>echo &quot;string lower&quot; . strtolower($str);echo &quot;&lt;br&gt;&quot;;</td>
</tr>
<tr>
<td>&lt;/form&gt;</td>
<td>echo &quot;string reverse&quot; . strrev($str);</td>
</tr>
<tr>
<td>&lt;/body&gt;</td>
<td>?&gt;</td>
</tr>
<tr>
<td>&lt;/html&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**Output:**

If input is Nihar
String length is 5
String upper NIHAR
String lower Nihar
String reverse rahiN
EXPERIMENT NO 10:

AIM OF EXPERIMENT: Write a Menu-Driven program to implement a calculator which performs only addition, subtraction, multiplication and division. The operation should happen based on the user choice. (use switch case)

OBJECTIVE:
Illustrate the use of switch statements.

THEORY (IF ANY):
Use the switch statement to select one of many blocks of code to be executed.
Syntax

```
switch (n)
{
case label1:
    code to be executed if n=label1;
    break;
    case label2:
    code to be executed if n=label2;
    break;
    default:
    code to be executed if n is different from both label1 and label2;
}
```

This is how it works: First we have a single expression n (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use break to prevent the code from running into the next case automatically. The default statement is used if no match is found.

Example

```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, or green!";
    }
?>
```

Output would be
Your favorite color is red!
```php
<?php
$ch=0;
i=0;
j=0;
do
{
    printf("Please enter your choice\n");
    printf("1: Multiply\n");
    printf("2: Divide\n");
    printf("3: Add\n");
    printf("4: Subtract\n");
    printf("5. Exit\n");
    $ch=fgets(STDIN);
    switch($ch)
    {
    case 1;
        printf("Please enter two nos\n");
        $i=fgets(STDIN);
        $j=fgets(STDIN);
        echo "Result is";
        echo ($i*$j);
        break;
    case 2;
        printf("Please enter two nos\n");
        $i=fgets(STDIN);
        $j=fgets(STDIN);
        echo "Result is";
        echo ($i/$j);
        break;
    case 3;
        printf("Please enter two nos\n");
        $i=fgets(STDIN);
        $j=fgets(STDIN);
        echo "Result is";
        echo ($i+$j);
        break;
    case 4;
        printf("Please enter two nos\n");
        $i=fgets(STDIN);
        $j=fgets(STDIN);
        echo "Result is";
        echo ($i-$j);
        break;
    case 5;
        printf("Exiting\n");
        break;
    default:
        echo "Invalid Choice";
    }
}
}while($ch!=5);
?>
```
OUTPUT:
php calc.php

Please enter your choice
1: Multiply
2: Divide
3: Add
4: Subtract
5. exit

Please enter two nos
3
4
Result is -1

Please enter your choice
1: Multiply
2: Divide
3: Add
4: Subtract
5. exit

Exiting
**EXPERIMENT NO: 11**

**AIM OF THE EXPERIMENT:** Write a function to swap two string values using call by value and call by references.

**OBJECTIVE:**
Illustrate the difference between call by value and call by reference

**THEORY (IF ANY):**
There is no reference sign on a function call - only on function definitions. Function definitions alone are enough to correctly pass the argument by reference.

```php
<?php
function foo(&$var)
{
    $var++;
}

$a=5;
foo($a);
// $a is 6 here
?>
```

**SOURCE CODE:**

```php
//Pass by value
<?php
$str1="Sahil";
$str2="Raheja";
echo "Swapping of two strings using Call by value";
echo "Before swapping strings are<br>"."String 1 = ". $str1."<br>String 2 = ". $str2."<br>
swap ($str1, $str2);
echo "Back from function :

After swapping strings are<br>"."String 1 = ". $str1."<br>String 2 = ". $str2;
function swap ($x, $y)
{
    $temp=$x;
    $x=$y;
    $y=$temp;
    echo "In function :

After swapping strings are<br>String 1 = ".$x."<br>String 2 = ".$y."<br>
}
?>
```

Run the program with url localhost/swap.php

**OUTPUT:**

//Pass by value
Swapping of two strings using Call by value
Before swapping strings are
String 1 = Sahil
String 2 = Raheja
In function:
After swapping strings are
String 1 = Raheja
String 2 = Sahil
Back from function:
After swapping strings are
String 1 = Sahil
String 2 = Raheja

**SOURCE CODE:**

```php
<?php
$str1="Sahil";
$str2="Raheja";
echo "<br>Swapping of two strings using Call by reference<br>";
echo "Before swapping strings are<br>"."String 1 = ". $str1."<br>String 2 = ". $str2."<br>
swap ($str1,$str2);
echo "Back from function:<br>";
echo "After swapping strings are<br>"."String 1 = ".$str1."<br>String 2 = ".$str2;
function swap(&$x,&$y)
{
    $temp=$x;
    $x=$y;
    $y=$temp;
    echo "In function:<br>";
    echo "After swapping strings are<br>String 1 = ".$x."<br>String 2 = ".$y."<br>
}
?>
```

Run the program with url localhost/swap1.php

**OUTPUT:**
//Pass by reference
Swapping of two strings using Call by reference
Before swapping strings are
String 1 = Sahil
String 2 = Raheja
In function:
After swapping strings are
String 1 = Raheja
String 2 = Sahil
Back from function:
After swapping strings are
String 1 = Raheja
String 2 = Sahil
EXPERIMENT NO: 12

AIM OF THE EXPERIMENT: Write a program that will accept an array of integers as input, and output an array where for each item in the source array, the new array will perform the following operations:

- For even numbers divide by 2
- For odd numbers multiply by 3

OBJECTIVE:
Array handling in php.

THEORY (IF ANY):

SOURCE CODE:

```php
<?php
$numbers=array(10,20,5,16,38,97,23,11,10,32,35,453,21,436,865);
$count1=count ($numbers);
for ($i=0; $i<$count1; $i=$i+1)
{
    if ($numbers [$i] %2==0)
        $numbers [$i] =$numbers [$i]/2;
    else if ($numbers [$i] %3==0)
        $numbers [$i] =$numbers[$i]*3;
}
echo "After operation array values are <br>
foreach ($numbers as $value)
echo $value."<br">;
?>
```

OUTPUT:
Output: After operation array values are
5
10
15
8
19
291
69
33
5
16
105
1359
63
218
2595
**EXPERIMENT NO: 13**

**AIM OF THE EXPERIMENT:** Create an associative array using the countries as keys, the cities as values and transform it into 2-dimensional array and display the data as a table.

**OBJECTIVE:**
Illustrate how to use associative arrays in PHP.

**THEORY (IF ANY):**
An associative array can be created using the array() language construct. It takes any number of comma-separated key => value pairs as arguments.
array(  
    key => value,  
    key2 => value2,  
    key3 => value3,  
    ...  
)
The comma after the last array element is optional and can be omitted. This is usually done for single-line arrays, i.e. array(1, 2) is preferred over array(1, 2, ).

For multi-line arrays on the other hand the trailing comma is commonly used, as it allows easier addition of new elements at the end.

**SOURCE CODE:**

```html
<html>
<body>
<?php
$a_array = array(  
    "India" => "NewDelhi",  
    "Afghanistan" => "Kabul",  
    "Australia" => "Canberra",  
    "Bangladesh" => "Dhaka",  
    "Brazil" => "Brasilia",  
    "Canada" => "Ottawa",  
    "China" => "Beijing",  
    "France" => "Paris",  
    "Japan" => "Tokyo",  
    "Thailand" => "Bangkok"  
);
$j=0;
foreach ($a_array as $x=>$x_value)  
{  
    $newarray[0][$j]=$x;  
    $newarray[1][$j]=$x_value;  
    $j=$j+1;  
}
echo "<table border=1>";
echo "<tr><th>Country</th><th>Capital</th></tr>";
for($k=0;$k<$j;$k++)  
{echo "<tr><td>".$newarray[0][$k]."</td>";  
  echo "<td>".$newarray[1][$k]."</td></tr>";  
}
echo "</table>";
?>
</body>
</html>
```
### OUTPUT:

<table>
<thead>
<tr>
<th>Country</th>
<th>Capital</th>
</tr>
</thead>
<tbody>
<tr>
<td>India</td>
<td>NewDelhi</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Kabul</td>
</tr>
<tr>
<td>Australia</td>
<td>Canberra</td>
</tr>
<tr>
<td>Bangladesh</td>
<td>Dhaka</td>
</tr>
<tr>
<td>Brazil</td>
<td>Brasilia</td>
</tr>
<tr>
<td>Canada</td>
<td>Ottawa</td>
</tr>
<tr>
<td>China</td>
<td>Beijing</td>
</tr>
<tr>
<td>France</td>
<td>Paris</td>
</tr>
<tr>
<td>Japan</td>
<td>Tokyo</td>
</tr>
<tr>
<td>Thailand</td>
<td>Bangkok</td>
</tr>
</tbody>
</table>
Experiment No: 14

Aim of the Experiment: Given two strings A and B, how would you find out if the characters in B were a subset of the characters in A?

Objective: 
Theory (if any):
Source Code:
Output:
**EXPERIMENT NO: 15**

**AIM OF THE EXPERIMENT:** Write a program that creates a file and writes contents to it and display it. Then append some data to it.

**OBJECTIVE:**
File handling using php.

**THEORY (if any):**
fopen — Opens file or URL
resource fopen ( string $filename , string $mode [, bool $use_include_path = false [, resource $context ]])

fopen() binds a named resource, specified by filename, to a stream. If filename is of the form "scheme://...", it is assumed to be a URL and PHP will search for a protocol handler (also known as a wrapper) for that scheme. If no wrappers for that protocol are registered, PHP will emit a notice to help you track potential problems in your script and then continue as though filename specifies a regular file.

If PHP has decided that filename specifies a local file, then it will try to open a stream on that file. The file must be accessible to PHP, so you need to ensure that the file access permissions allow this access. If you have enabled safe mode, or open_basedir further restrictions may apply.

If PHP has decided that filename specifies a registered protocol, and that protocol is registered as a network URL, PHP will check to make sure that allow_url_fopen is enabled. If it is switched off, PHP will emit a warning and the fopen call will fail.

```
$handle = fopen("c:\folder\resource.txt", "r");
```

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'r'</td>
<td>Open for reading only; place the file pointer at the beginning of the file.</td>
</tr>
<tr>
<td>'r+'</td>
<td>Open for reading and writing; place the file pointer at the beginning of the file.</td>
</tr>
<tr>
<td>'w'</td>
<td>Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.</td>
</tr>
<tr>
<td>'w+'</td>
<td>Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.</td>
</tr>
<tr>
<td>'a'</td>
<td>Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.</td>
</tr>
<tr>
<td>'a+'</td>
<td>Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.</td>
</tr>
<tr>
<td>'x'</td>
<td>Create and open for writing only; place the file pointer at the beginning of the file. If the file already exists, the fopen() call will fail by returning FALSE and generating an error of level E_WARNING. If the file does not exist, attempt to create it. This is equivalent to specifying O_EXCL</td>
</tr>
<tr>
<td>'x+'</td>
<td>Create and open for reading and writing; otherwise it has the same behavior as 'x'.</td>
</tr>
<tr>
<td>'c'</td>
<td>Open the file for writing only. If the file does not exist, it is created. If it exists, it is neither truncated (as opposed to 'w'), nor the call to this function fails (as is the case with 'x'). The file pointer is positioned on the beginning of the file. This may be useful if it's desired to get an advisory lock (see flock()) before attempting to modify the file, as using 'w' could truncate the file before the lock was obtained (if truncation is desired, ftruncate() can be used after the lock is requested).</td>
</tr>
</tbody>
</table>
Open the file for reading and writing; otherwise it has the same behavior as 'c'.

Append to a file
If we want to add on to a file we need to open it up in append mode. The code below does just that.

```php
<?php
$my_file = 'nihar.txt';
$handle = fopen($my_file, 'a') or die('Cannot open file: '.$my_file);
$myFile = "testFile.txt";
$fh = fopen($myFile, 'a');

If we were to write to the file it would begin writing data at the end of the file.

SOURCE CODE:
```
```
Experiment No: 16

Aim of the Experiment: Create a login form with two text fields called "login" and "password". When user enters “Galgotias” as a user name and “university” as a password it should be redirected to a Welcome.HTML page or to Sorry.HTML in case of wrong username/password.

Objective:

Theory (if any):

Source Code:

```html
Login.html
<html>
<head>
    <title>PHP Login</title>
</head>
<body bgcolor="#FFFFCC">
    <center>
        <h3>PHP LOGIN</h3>
        <hr />
        <form method="post" action="login.php">
            <table border="0">
                <tr>
                    <td><b>User</b></td>
                    <td><input type="text" name="userid">
                </tr>
                <tr>
                    <td><b>Password</b></td>
                    <td><input name="password" type="password"></input></td>
                </tr>
                <tr>
                    <td><input type="submit" value="Submit"/>
                    <td><input type="reset" value="Reset"/>
                </tr>
            </table>
        </form>
    </center>
</body>
</html>

//login.php
<html>
<body bgcolor="#FFFFCC">
    <center>
        <?php
        $f_usr= $_POST["userid" ];
        $f_pswd= $_POST["password" ];
        $con=mysql_connect("localhost","root" ," ");
        if(! $con) {
            die('Connection Failed'.mysql_error());
        }
        
```
```php
mysql_select_db("abc",$con);
$result=mysql_query("select * from login1");
while($row=mysql_fetch_array($result))
{
    if($row["username"]==$f_usr && $row["password"]==$f_pswd)
    {
        echo"Welcome : $f_usr"
    }
    else
    {
        echo"Sorry $f_usr, your user id or password is not correct"
        include("login1.html")
    }
}
?>
</h1>
</center>
</body>
</html>
```

**OUTPUT:**
**Experiment No: 17**

**Aim of the Experiment:** Write a PHP program using Java Script to convert the decimal number to its binary equivalent. You must use a form to accept the number from the user.

**Objective:**

**Theory (if any):**

**Source Code:**

```html
<html>
<body>
</b>Insert Decimal Number: </b><input type = "text" name = "deci" id = "deci" size = "15" maxlength = "15" />
<input type="button" value="Convert!" onclick="dec2bin()" />
<br><br>
<div id = "result"></div>
<script type = "text/javascript">
function dec2bin() {
    var x = document.getElementById("deci").value;
    if (/(^[0-9]*)\d\>/.test(x) || x == "") {
        alert ("You must enter an integer decimal number!");
        document.getElementById("deci").value = ";
        document.getElementById("deci").focus();
        return false;
    }
    x = parseInt(x);
    var bin = x.toString(2);
    var hex = x.toString(16).toUpperCase();
    var octal = x.toString(8);
    var figs = "The binary representation of " + x + " is " + bin + "<br>
    figs = figs + "The hexadecimal representation of " + x + " is " + hex + "<br>
    figs = figs + "The octal representation of " + x + " is " + octal + "<br>";
    document.getElementById("result").innerHTML = figs;
}
</script>
</body>
</html>
```

**Output:**
Insert Decimal Number: 13

The binary representation of 13 is 1101
The hexadecimal representation of 13 is D
The octal representation of 13 is 15
**Experiment No: 18**

**Aim of the Experiment:** Write a PHP code that define class Student with attributes RollNo, Name, Branch, and Year, create 3 instances of it, sets the values of each instance appropriately and print the values of all attributes.

**Objective:**
Illustrate object oriented programming through PHP.

**Theory (if any):**

**Source Code:**
```php
<?php
class Employee
{
    public $RollNo;
    public $Branch;
    public $Name;
    public $Year;
    function set($RollNo, $Branch, $Name, $Year)
    {
        $this->RollNo = $RollNo;
        $this->Branch = $Branch;
        $this->Name = $Name;
        $this->Year = $Year;
    }
    function getRollNo()
    { return $this->RollNo; }
    function getBranch()
    { return $this->Branch; }
    function getName()
    { return $this->Name; }
    function getYear()
    { return $this->Year; }
    function __construct()
    {}
    function __destruct()
    {}
}
$obj[1] = new Employee();
$obj[2] = new Employee();
$obj[3] = new Employee();
$obj[1]->set(1001, "CSE", "amit", 2012);
$obj[2]->set(1002, "CSE", "prashant", 2011);
$obj[3]->set(1003, "ECE", "sachin", 2012);
for($j=1; $j<4; $j++)
{
    echo $obj[$j]->getRollNo();
    echo $obj[$j]->getBranch();
    echo $obj[$j]->getName();
    echo $obj[$j]->getYear()."\n";
}
?>
```
OUTPUT:

```
1001CSEAmrit2012
1002CSEPrahashant2011
1003CEEsachin2012
Press any key to continue . . .
```
**Experiment No: 19**

**Aim of the Experiment:** Write a function `calculateAverage()` which takes four int arguments which are marks for four courses in the semester and returns their average as a float. The `calculateAverage()` function should take only valid range for marks which is between 0 - 100. If the marks are out of range is should throw an `OutOfRangeException` and handles it.

**Objective:**

**Theory (if any):**

**Source Code:**

**Output:**
EXPERIMENT NO: 20

AIM OF THE EXPERIMENT: Create a form with a text box asking to enter your favorite city with a submit button when the user enters the city and clicks the submits button another php page should be opened displaying “Welcome to the city”.

OBJECTIVE:
THEORY (IF ANY):
SOURCE CODE:
OUTPUT: