Web Based Application Development With PHP

Sub Code : 22619

EDITION : 2020

INCLUDES-I SCHEME PATTERN
SAMPLE PAPERS

MSBTE I SCHEME PATTERN
T. Y. DIPLOMA SEM VI
COMPUTER ENGINEERING GROUP
(CO/CM/IF/CW)

- INCLUDES LABORATORY PROGRAMS

TECHNICAL®

SUBJECT CODE : 22619

As per Revised Syllabus of

MSBTE - I SCHEME

S.Y. Diploma Semester - VI
Computer Engineering Group
(CO / CM / IF / CW)

WEB BASED APPLICATION DEVELOPMENT WITH PHP

Anuradha A. Puntambekar
M.E. (Computer)
Formerly Assistant Professor in
P.E.S. Modern College of Engineering,
Pune

TECHNICAL PUBLICATIONS
SINCE 1993
An Up-Thrust for Knowledge

WEB BASED APPLICATION DEVELOPMENT WITH PHP

Subject Code : 22619

S.Y. Diploma Semester - VI
Computer Engineering Group (CO / CM / IF / CW)

First Edition : January 2020

Published by :

Amit Residency, Office No.1, 412, Shaniwar Peth, Pune - 411030, M.S. INDIA
Ph.: +91-020-24495496/97, Telefax : +91-020-24495497
Email : sales@technicalpublications.org Website : www.technicalpublications.org

Printer :
Yogiraj Printers & Binders
Sr.No. 10/1A,
Ghule Industrial Estate, Nanded Village Road,
Tal. - Haveli, Dist. - Pune - 411041.

Price : ₹ 110/-

ISBN 978-93-89750-01-0

MSBTE I

PREFACE

The importance of **Web Based Application Development with PHP** is well known in various engineering fields. Overwhelming response to my books on various subjects inspired me to write this book. The book is structured to cover the key aspects of the subject **Web Based Application Development with PHP**.

The book uses plain, lucid language to explain fundamentals of this subject. The book provides logical method of explaining various complicated concepts and stepwise methods to explain the important topics. Each chapter is well supported with necessary illustrations and practical examples. All the chapters in the book are arranged in a proper sequence that permits each topic to build upon earlier studies. All care has been taken to make students comfortable in understanding the basic concepts of the subject.

Representative questions have been added at the end of each section to help the students in picking important points from that section.

The book not only covers the entire scope of the subject but explains the philosophy of the subject. This makes the understanding of this subject more clear and makes it more interesting. The book will be very useful not only to the students but also to the subject teachers. The students have to omit nothing and possibly have to cover nothing more.

I wish to express my profound thanks to all those who helped in making this book a reality. Much needed moral support and encouragement is provided on numerous occasions by my whole family. I wish to thank the **Publisher** and the entire team of **Technical Publications** who have taken immense pain to get this book in time with quality printing.

Any suggestion for the improvement of the book will be acknowledged and well appreciated.

Author
A. A. Puntambekar

Dedicated to God...

SYLLABUS

Web Based Application Development with PHP (22619)

Teaching Scheme			Credit (L+T+P)	Examination Scheme											
				Theory						Practical					
L	T	P		Paper Hrs.	ESE		PA		Total		ESE		PA		Total	
					Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
3	-	2	5	3	70	28	30*	00	100	40	25@	10	25	10	50	20

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub - topics
Unit - I Expressions and control statements in PHP	1a. Write simple PHP program to solve the given expression. 1b. Use relevant decision making control statement to solve the given problem. 1c. Solve the given iterative problem using relevant loop statement.	1.1 History and Advantages of PHP, Syntax of PHP. 1.2 Variables, Data types, Expressions and operators, constants. 1.3 Decision making Control statements - if, if-else, nested if, switch, break and continue statement. 1.4 Loop control structures - while, do - while, for and foreach.
Unit - II Arrays, Functions and Graphics	2a. Manipulate the given type of arrays to get the desired result. 2b. Apply implode, explode functions on the given array. 2c. Apply the given string functions on the character array. 2d. Scale the given image using graphics concepts/functions.	2.1 Creating and Manipulating Array, Types of Arrays - Indexed, Associative and Multi-dimensional arrays. 2.2 Extracting data from arrays, implode, explode, and array flip. 2.3 Traversing Arrays. 2.4 Function and its types - User defined function, Variable function and Anonymous function. 2.5 Operations on String and String functions : str_word_count(),strlen(),str rev(), strpos(), str_replace(), ucwords(), strtoupper(), strtolower(), strcmp(). 2.6 Basic Graphics Concepts, Creating Images, Images with text, Scaling Images, Creation of PDF document.

Unit - III **Apply Object Oriented Concepts in PHP**	3a. Write constructor and destructor functions for the given problem in PHP. 3b. Implement inheritance to extend the given base class. 3c. Use overloading / overriding to solve the given problem. 3d. Clone the given object.	3.1 Creating Classes and Objects 3.2 Constructor and Destructor 3.3 Inheritance, Overloading and Overriding, Cloning Object. 3.4 Introspection, Serialization
Unit - IV **Creating and validating forms**	4a. Use the relevant form controls to get user's input. 4b. Design web pages using multiple Forms for the given problem. 4c. Apply the given validation rules on form. 4d. Set/ modify/ delete cookies using cookies attributes. 4e. Manage the given session using session variables.	4.1 Creating a webpage using GUI Components, Browser Role-GET and POST methods, Server Role 4.2 Form controls : text box, text area, radio button, check box, list, buttons. 4.3 Working with multiple forms : - A web page having many forms. - A form having multiple submit buttons. 4.4 Web page validation. 4.5 Cookies - Use of cookies, Attributes of cookies, create cookies, modify cookies value, and delete cookies. 4.6 Session- Use of session, Start session, get session variables, destroy session. 4.7 Sending E-mail.
Unit - V **Database Operations**	5a. Create database for the given problem using PHP script. 5b. Insert data in the given database using PHP script. 5c. Apply the specified update operation in database record using PHP script. 5d. Delete the given record from the database using PHP script.	5.1 Introduction to MySQL - Create a database. 5.2 Connecting to a MySQL database : MySQL database server from PHP. 5.3 Database Operations : Insert data, Retrieving the Query result. 5.4 Update and delete operations on table data.

TABLE OF CONTENTS

Unit - V

UNIT - I

1 Expressions and Control Statements in PHP

1.1 History and Advantages of PHP

History

- PHP was developed in 1994 by Apache group.
- PHP stands for PHP : **H**ypertext **P**reprocessor.
- PHP is a server-side scripting language. It is mainly used for form handling and database access.
- It is free to download and use.

Advantages

Various advantages of PHP are

1. It is very **simple** and **easy to learn** and use. It is **widely used** scripting language.
2. It is an **interpreted language** and there is **no need for compilation**.
3. It is **open source** scripting language. That means you can **freely download** and use php.
4. It is **platform independent.** That means , PHP code can run on various platforms such as Windows, Linux, Mac OS.
5. The php code can be **directly integrated with HTML**.
6. It is designed to **support dynamic web applications**.
7. This is **reliable**, **efficient** and **flexible** scripting language.
8. It provide **support** for – file system, managing user sessions, cookies, E-mail management, execute the system commands, create directories and so on.

1.1.1 Syntax of PHP

- PHP code can be embedded in the HTML document. The code must be enclosed within <**?php** and >
- If the PHP script is stored in some another file and if it needs to be referred then **include** construct is used. For instance:

```
Include("myfile.html")
```

- The variable names in PHP begin with the $ sign.
- Following are some reserved keywords that are used in PHP.

and	default	false	if	or	this
break	do	For	include	require	true
case	else	foreach	list	return	var
class	elseif	function	new	static	virtual

continue	extends	gloabal	not	switch	while
					xor

- The **comments** in PHP can be #,//, /* …*/
- The PHP statements are terminated by semicolon.

How to write and execute PHP documents ?

Open some suitable text editor like Notepad and type the following code. Save your code by the extension **.php.**

It is expected that the PHP code must be stored in **htdocs** folder of Apache.

As I have installed **xampp**, I have got the directory c:\xampp\htdocs. I have created a folder named php-examples inside the **htdocs** and stored all my PHP documents in that folder.

Hence when I want to get the output of the PHP code I always give the URL

http://localhost/php-examples/programmName.php

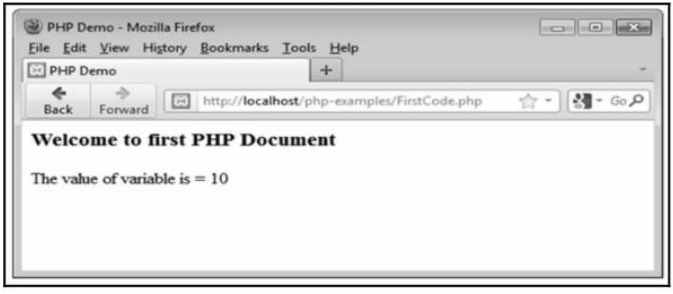
The http://localhost refers to the path **c:\xampp\htdocs**

Following is the first example of PHP script

PHP Document[FirstCode.php]

```
<html>
 <head>
  <title> PHP Demo </title>
 </head>
 <body>
  <?php
  $i=10;
  echo "<h3>Welcome to first PHP Document</h3>";
  echo "The value of variable is = $i";
  ?>
 </body>
</html>
```

Output



1.1.2 How to Write and Execute PHP Program ?

The most popular way of installing PHP is using **XAMPP**.

XAMPP is a free distribution package that makes it easy to install Apache Web Server, MySQL, PHP, PEAR. Here in XAMPP (X stands for any OS) or WAMPP (W stands for Windows OS).

Step 1 : Go to the site **https://www.apachefriends.org/index.html**

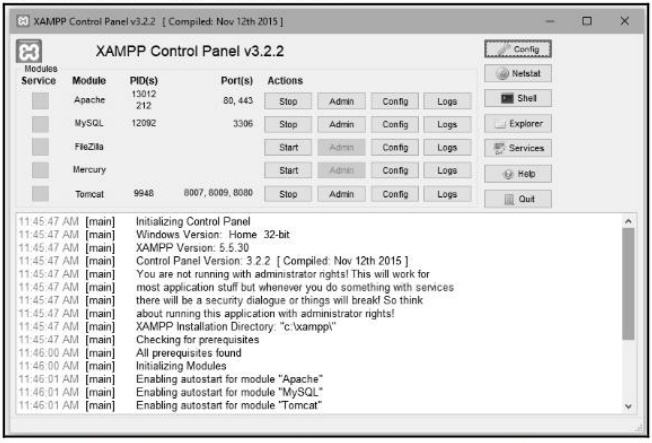
Step 2 : Click on Download XAMPP for Windows or Linux depending upon your operating system.

Step 3 : When prompted for the download, click "Save" and wait for your download to finish.

Step 4 : Install the program, and click on "Run." Accept default settings by clicking Next button. Finally you will get installation completion message.

Step 5 : On your drive, the XAMPP folder will be created. Click on xampp_start file, this will enable to start Apache, MySQL and Tomcat start.

Step 6 : The control panel for XAMPP will look like this,



Step 7 : Write a PHP script and save it in **C:\XAMPP\htdocs\php-examples** folder by giving the filename and extension as .php

Step 8 : Open the web browser and type **http://localhost/php-examples/yourfilename.php**

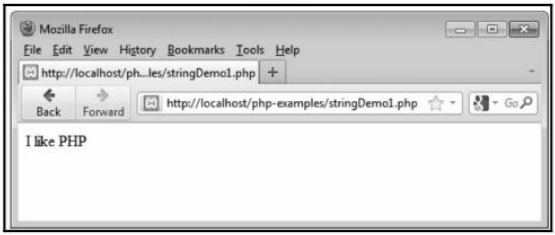
Step 9 : The web application will be executed within your web browser.

For example

PHP Script[stringDemo1.php]

```
<?php
$s="I like PHP";
echo $s;
?>
```

Output



Review Questions

1. *What is PHP ? Give general syntax of PHP.*
2. *Give the advantages of PHP.*

1.2 Variables, Data types, Expressions, Operators and Constants

1.2.1 Variables

- Variables are the entities that are used for storing the values.
- PHP is a dynamically typed language. That is PHP has no type declaration.
- The value can be assigned to the variable in following manner -
  $variable_name=value
- If the value is not assigned to the variable then by default the value is NULL. The unsigned variables are called **unbound variable.**
- If the unbound variable is used in the expression then its NULL value is converted to the value 0.
- Following are some rules that must be followed while using the variables -
  1. The variable must start with letter or underscore _, but it should not begin with a number.
  2. It consists of alphanumeric characters or underscore.
  3. There should not be space in the name of the variable.
  4. While assigning the values to the variable the variable must start with the $. For example
     $marks=100;

1.2.2 Data Types

There are four scalar types that are used in PHP and those are Integer, Boolean, Double and String. Let us discuss each one by one.

Integer Type

- For displaying the integer value the Integer type is used.
- It is similar to the **long** data type in C.
- The size is 32 bit.

Double Type

- For displaying the real values the double data type is used.

- It includes the numbers with decimal point, exponentiation or both. The exponent can be represented by E or e followed by integer literal.
- It is not compulsory to have digits before and after the decimal point. For instance .123 or 123. is allowed in PHP.

String Type

- There is no character data type in PHP. If the character has to be represented then it is represented using the string type itself; but in this case the string is considered to be of length 1.
- The string literals can be defined using either single or double quotes.
- In single quotes the escape sequence or the values of the literals can not be recognized by PHP but in double quotes the escape sequences can be recognized. For example

'The total marks are= $marks'

will be typed as it is but

"The total marks are= $marks"

will display the value of $marks variable.

Boolean Type

- There are only two types of values that can be defined by the Boolean type and those are TRUE and FALSE.
- If Boolean values are used in context of integer type variable then TRUE will be interpreted as 1 and FALSE will be interpreted as 0.
- If Boolean values are used in context of double type then the FALSE will be interpreted as 0.0.

1.2.3 Constants

- Constant is an identifier that contains some value.
- Once the constant value is assigned to this identifier it does not get changed.
- Constant is case sensitive by default.
- Generally the constant identifiers are specified in upper case.
- The valid constant name must start with letter or underscore. It may then followed by the digits.
- Using define function we can assign value to the constant. The first parameter in define function is the name of the constant and the second parameter is the value which is to be assigned.

For example :

ConstDemo.php

```
<?php
// Valid constant names
define("MYVALUE","10");
echo MYVALUE;
// Invalid constant names
define("1MYVALUE","something");
echo 1MYVALUE;
?>
```

These statements will cause error. If we remove these statements then we will get **10** as an output.

1.2.4 Operators

1. Arithmetic Operators and Operations

- PHP supports the collection of arithmetic operators such as +,-,/,*,%,++ and – with their usual meaning.
- While using the arithmetic operators if both the operands are integer then the result will be integer itself.
- If either of the two operands is double then the result will be double.
- PHP has large number of predefined functions. Some of these functions are enlisted in the following table -

Function	Purpose
floor	The largest integer less than or equal to the parameter is returned For example floor(4.9) will return 4,
ceil	The smallest integer less than or equal to the parameter is returned For example ceil(4.9) will return 5,
round	Nearest integer is returned.
abs	Returns the absolute value of the parameter.
min	It returns the smaller element .
max	It returns the larger element .

2. Relational Operators

- There are eight relational operators used in PHP.
- These are <,>,<=,>=,!=,== has their usual meaning. These are six traditional operators.
- The operator === is used in PHP. It returns true if both operands that are using === have same type and have same value.
- The operator !== is opposite of ===.
- If one of the operand in the six operators is not same then the coercion will occur automatically.

3. Boolean Operators

- The Boolean operators are

Operator	Meaning
and &&	The binary AND operation is performed.
or ||	The binary OR operation is performed.
xor	The XOR operation will be performed.

4. Autoincrement or Autodecrement Operator

The unary ++ or -- operators are used as autoincrement or autodecrement operators. For example

Name	Operator	Value Returned
Post-increment	$a++	$a
Pre-increment	++$a	$a+1
Post-decrement	$a--	$a-1

5. Bitwise Operators

Following are bitwise operators used in PHP.

Operator	Name	Example	Result
&	And	$x & $y	Bits that are set in both $x and $y are set.
|	Or	$x | $y	Bits that are set in either $x or $y are set.
^	Xor	$x ^ $y	Bits that are set in $x or $y but not both are set.
~	Not	~$x	Bits that are set in $x are not set, and vice versa.
<<	Shift left	$x << $y	Shift the bits of $x $y steps to the left. Another purpose of this operator can also be "multiply by two".
>>	Shift right	$x >> $y	Shift the bits of $x $y steps to the right. Another purpose of this operator can also be "divide by two".

A bit (Binary digit) is the basic unit of information stored in the computing system that exists in two possible states, represented as ON or OFF. In a computer system, the ON state considered as 1 and OFF state considered as 0.

The truth table for bitwise operations is as given below.

a	b	a|b	a&b	a^b
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

For example

Consider $a=40 and $b=80. Binary form of these values is given below.

$a = 00101000

$b= 01010000

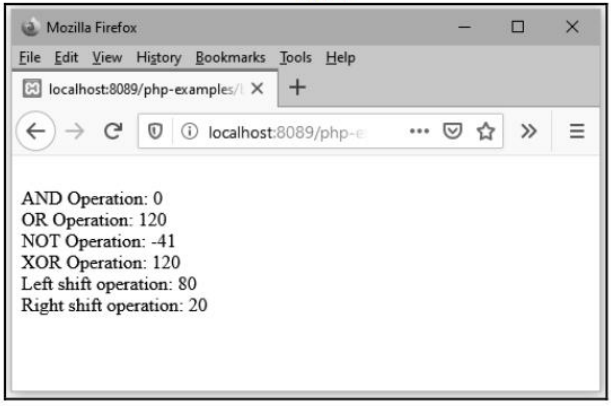
Operation	Result in Binary	Result in Decimal
a&b	00000000	0
a|b	01111000	120

~a	Value of 40 in binary is 00000000000000000000000000000000 00000000000000000000101000000000. So, all 0's are converted into 1's in bit wise NOT operation. Hence result is 11111111111111111111111111111111111111111111111111111111111010111	- 41
a^b	01111000	120
a<<1	01010000	80
a>>1	00010100	20

Programming Example

```
<?php
$a = 40;
$b = 80;
$AND_opr = ($a & $b);
$OR_opr = ($a | $b);
$NOT_opr = (~$a);
$XOR_opr = ($a ^ $b);
echo "<br/>AND Operation: ".$AND_opr;
echo "<br/>OR Operation: ".$OR_opr;
echo "<br/>NOT Operation: ".$NOT_opr;
echo "<br/>XOR Operation: ".$XOR_opr;
echo "<br/> Left shift operation: ".($a<<1);
echo "<br/> Right shift operation: ".($a>>1);
?>
```

Output



6. Assignment Operator

The assignment operator is used to assign the values to variables. Following is a list of assignment operators.

Operator	Example and Meaning
=	This is an operator using which values is assigned to a variable a = 5
+=	a += 5 means a = a + 5
-=	a -= 5 means a = a - 5

Similarly *=, /= operators are used for performing arithmetic multiplication and division operation.

1.2.5 Expressions

- Expressions are most important building blocks in PHP.
- Expression can be defined as an entity that has some value.
- For example

```
$a=100;
```

is an expression

- The complex form of expression is a function. For example

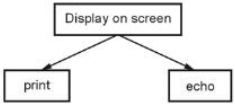
```
?>php
function fun()
{
    return 1;
}
$val=fun();
```

- We can form an expression using the post-increment and pre-increment operators.
- For example -

```
$a++ or ++$a
```

1.2.6 Displaying Messages on Screen

- In PHP the output can be displayed using two statements :



Display using Print

- The **print** function is used to create simple unformatted output. For example: The string can be displayed as follows

```
print "I am proud of my <b>country</b>"
```

The numeric value can also be displayed using the **print**. For example -

```
print(100);
```

It will display the output as 100.

- PHP also makes use of the **printf** function used in C. For example

  printf("The student %d has %f marks",$roll_no,$marks);

- Following is a simple PHP document which makes use of the statements for displaying the output.

PHP Document[OutputDemo.php]

```
<html>
<head>
 <title> Output Demo</title>
</head>
<body>
 <?php
 print "<h2>Welcome to my Website </h2>";
 print "<hr/>";
 $roll_no=1;
 $name="AAA";
 printf("<b>The roll number: %d</b>",$roll_no);
      print "<br/><b>";
 printf("The name: %s",$name);
 print "</b>";
?>
</body>
</html>
```

Output

Display using echo

- The echo is a simple statement that can be used with or without parenthesis.
- For example –

  echo "I am proud of my country"

- The echo statement can display the text and variable value as follows –

  $a=10;
  echo "a = ". $a;

Difference between echo and print

echo in PHP	print in PHP
The echo can output one or more strings.	Print can only output one string.
echo is faster than print.	print is slower than echo
It does not return any value	Print always returns 1.

Review Questions

1. *List and explain various data types in PHP.*
2. *Explain different types of operators used in PHP.*
3. *Explain the term expression with suitable example.*

1.3 Decision Making Control Statements

Decision making control statements decide the order of execution of the statements based on certain conditions. Various types of decision making control statements are -

1. The if Statement

Syntax

```
if(expression)
{
   statement inside;
}
   statement outside;
```

Example

```
<?php
$a = 80;
$b = 40;
if($a>$b)
 echo "a is greater than b";
?>
```

2. The if-else Statement

Syntax

```
if(expression)
{
   Statements;
}
else
{
        Statements;
}
```

Example

```
<?php
$a = 80;
$b = 40;
if($a>$b)
 echo "a is greater than b";
else
 echo "b is greater than a";
?>
```

3. The if-elseif... Statement

This type of statement combines multiple if…else statements

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
}
```

Example

```
<?php
$num = 100;
if($num>100)
 echo "Number is greater than 100";
elseif($num<100)
 echo "Number is less than 100";
else
 echo "Number is equal to 100";
?>
```

Ex. 1.3.1 : Write a PHP program to find the largest number among three numbers.

Sol. :

```
<html>
<head>
 <title>Selection Demo</title>
</head>
<body>
 <?php
 print "<h2>Selection Statement </h2>";
 $a=10;
 $b=20;
 $c=30;
 if($a>$b)
  if($a>$c)
      print "<b> <I>a is the largest number </I></b>";
     else
      print "<b><I> c is the largest number</I> </b>";
 else
  if($b>$c)
      print "<b><I>b is the largest number</I>  </b>";
      else
      print "<b> <I>c is the largest number</I> </b>";
?>
</body>
</html>
```

Output

4. The switch Statement

The switch case statement is used as an alternative to if…elseif…else statement.

Syntax

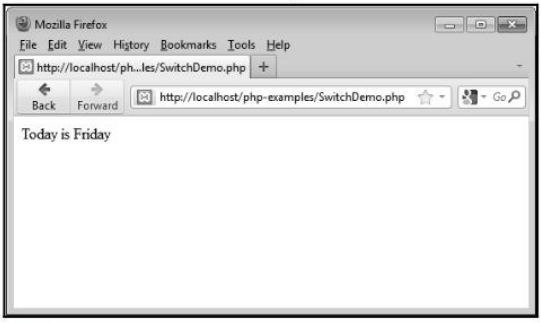
```
switch(n){
  case label1:
    // Code to be executed if n=label1
    break;
  case label2:
    // Code to be executed if n=label2
    break;
  ...
  default:
    // Code to be executed if n is different from all labels
}
```

Example

```
<?php
$today = getdate();
switch($today['weekday'])
{
 case "Monday":print "Today is Monday";
      break;
 case "Tuesday":print "Today is Tuesday ";
      break;
 case "Wednesday":print "Today is Wednesday ";
      break;
 case "Thursday":print "Today is Thursday";
      break;
 case "Friday":print "Today is Friday";
      break;
 case "Saturday":print "Today is Saturday";
```

```
        break;
 case "Sunday":print "Today is Sunday ";
        break;
   default: print "Invalid input";
}
?>
```

Output



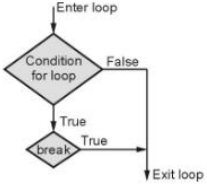
5. break and continue

The break Statement

- The break statement is used to transfer the control to the end of the loop.
- When break statement is applied then loop gets terminates and the control goes to the next line pointing after loop body.
- The flowchart for break statement is,

Syntax

break



For example

```
$i = 0;
for ($i = 0;$i <= 7;$i++)
{
   if ($i==3)
   {
      break;
   }
```

```
    echo $i;
    echo "<br />";
}
```

Output

0

1

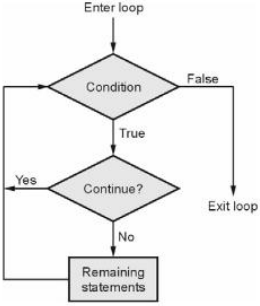
2

The continue Statement

- The continue statement is used to skip some statements inside the loop. The continue statement is used with decision making statement such as if...else.
- The continue statement forces to execute the next iteration of the loop to execute.
- The flowchart for continue statement is,

Syntax

Continue



For example

```
$i = 0;
for ($i = 0;$i <= 7;$i++)
{
    if ($i==3)
    {
        continue;
    }
    echo $i;
    echo "<br />";
}
```

Output

0

1

2

4

5
6
7

Difference between break and continue

Sr. No.	break	continue
1.	This statement terminates the execution of remaining iteration of the loop.	It terminates only the current iteration of the loop.
2.	It causes early termination of the entire loop.	It causes early execution of the next iteration.

1.4 Loop Control Structures

Loop control structures are used to execute the certain group of statements repetitively for certain number of times.

1. The while Loop

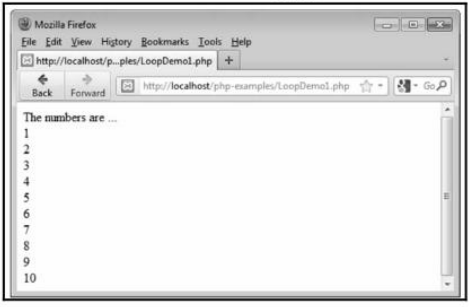
Syntax

```
while(condition){
   // Code to be executed
}
```

Example

```
<?php
$i=1;
print "The numbers are ...";
print "<br/>";
while($i<=10)
{
 print $i;
 print "<br/>";
 $i++;
}
?>
```

Output



2. The do...while Loop

In this type of looping statement the block of code executed once and then condition is evaluated. If the condition is true the statement is repeated as long as the specified condition is true.

Syntax

```
do{
   // Code to be executed
} while(condition);
```

Example

```
<?php
$i=1;
print "The numbers are ...";
print "<br/>";
do
{
 print $i;
 print "<br/>";
 $i++;
} while($i<=10);
?>
```

Output

The output will be same as above.

3. The for Loop

The for loop repeats a block of code as long as a certain condition is met. It is typically used to execute a block of code for certain number of times.

Syntax

```
for(initialization; condition; increment){
   // Code to be executed
}
```

Example

```
<?php
print "The numbers are ...";
print "<br/>";
for($i=1;$i<n)
{
 print $i;
 print "<br/>";
 $i++;
}
?>
```

Output

The output will be same as above.

4. The foreach Statement

The foreach statement is normally used to iterate through all the elements of array. It is discussed in section ** when we discuss the concept of arrays.

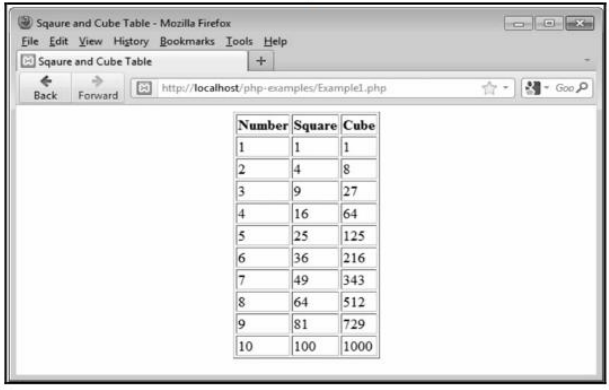
1.5 Programming Examples based on Control Structures

Ex. 1.5.1 : ***Write a PHP script to display the squares and cubes of 1 to 10 numbers.***

Sol. :

```
<html>
<head>
<title> Sqaure and Cube Table </title>

</head>
<body>
<center>
<?php
print "<table border =1>";
print "<tr>";
print "<th>Number</th>";
print "<th>Square</th>";
print "<th>Cube</th>";
print "</tr>";
for($i=1;$i<=10;$i++)
{
print "<tr>";
print "<td>$i";
print "</td>";
print "<td>";
print $i*$i;
print "</td>";
print "<td>";
print pow($i,3);
print "</td>";
print "</tr>";
}
print "</table>";
?>
</center>
</body>
</html>
```

Output



Number	Square	Cube
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

Ex. 1.5.2 : ***Write a PHP script to compute the sum and average of N numbers.***

Sol. :

```
<html>
<head>
<title> Sum and Average </title>
</head>
<body>
```

```
<center>
<?php
$sum=0;
for($i=1;$i<=10;$i++)
{
 $sum+=$i;
}
$avg=$sum/10;
print "The Sum is:  $sum";
print "<br/>";
print "the Average is: $avg";
?>
</center>
</body>
</html>
```

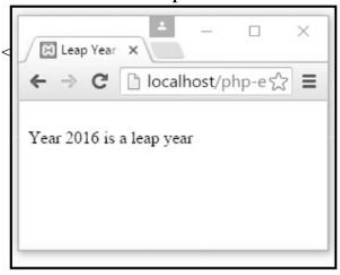
Output

Ex. 1.5.3 : ***Write PHP programs to print whether current year is leap year or not.***

Sol. :

```
<html>
<head>
<title>Leap Year Demo</title>
<body>
<?php
 $year=2016;
 print "<br/>";
 if($year%4==1)
  {  printf("Year %d is not a leap year",$year); }
 else
  {  printf("Year %d is a leap year",$year); }
?>
</body>
</html>
```

Output



Ex. 1.5.4 : ***Write PHP programs to print whether given number is odd or even.***

Sol. :

```
<html>
<head>
<title>Even Odd Demo</title>
</head>
<body>
<?php
for($i=1;$i<=10;$i++)
{
 $num=$i;
 print "<br/>";
 if($num%2==1)
  {  printf("Number %d is Odd",$num); }
 else
  {  printf("Number %d is Even",$num); }
}
```

Output

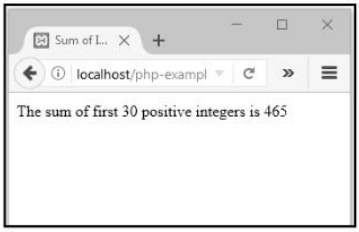
```
?>
</body>
</html>
```

Ex. 1.5.5 : ***Write PHP script to compute the sum of positive integers upto 30 using do-while statement.***

Sol. :

```
<html>
<head>
<title>Sum of Integers</title>
</head>
<body>
<?php
$sum=0;
$i=1;
do
{
 $sum=$sum+$i;
 $i++;
}while($i<=30);
 printf("The sum of first 30 positive integers is %d ",$sum);
?>
</body>
</html>
```

Output

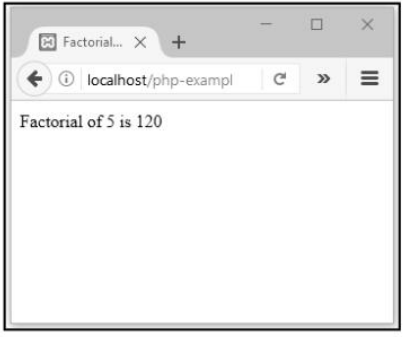


Ex. 1.5.6 : ***Write PHP script to compute factorial of 'n' using while or for loop construct.***

Sol. :

```
<html>
<head>
<title>Factorial Program</title>
</head>
<body>
<?php
$n = 5;
$factorial = 1;
for ($i=$n; $i>=1; $i--)
{
 $factorial = $factorial * $i;
}
echo "Factorial of $n is $factorial";
?>
</body>
</html>
```

Output

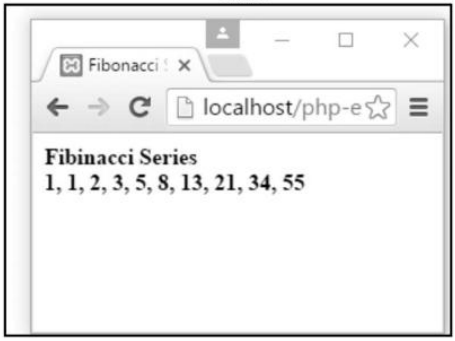


Ex. 1.5.7 : ***Write PHP script to display Fibonacci of length 10.***

Sol. :

```
<html>
<head>
<title>Fibonacci Series</title>
</head>
<body>
<?php
 $i=1;
 $j=1;
 print "<b>Fibinacci Series<br/>";
 printf("%d, %d",$i,$j);
 for($count=1;$count<9;$count++)
 {
  $k=$i+$j;
  $i=$j;
  $j=$k;
  printf(", %d",$k);
 }
?>
</body>
</html>
```

Output

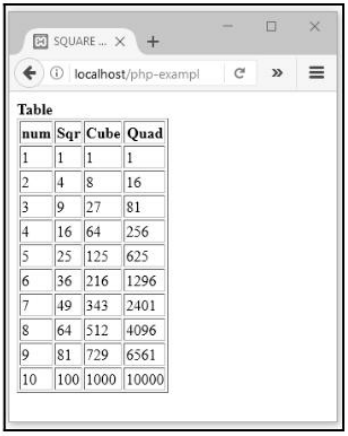


Ex. 1.5.8 : ***Construct a PHP script to compute the squareRoot, Square, Cube and Quad of 10 numbers.***

Sol. :

```
<html>
<head>
<title>SQUARE CUBE QUAD DEMO</title>
</head>
<body>
<?php

 print "<b>Table<br/>";
 print "<table border='1'>";
 print "<tr><th>num</th><th>Sqr</th><th>Cube</th><th>Quad</th></tr>";
 for($count=1;$count<=10;$count++)
 {
$sq=$count*$count;
$cube=$count*$count*$count;
$quad=$count*$count*$count*$count;
     print "<tr><td>$count</td><td>$sq</td><td>$cube</td><td>$quad</td></tr>";
 }
 print "</table>";
?>
</body>
</html>
```

Output

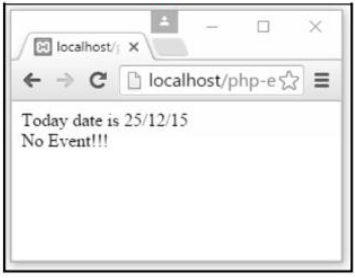


Ex. 1.5.9 : ***With the use of PHP, switch case and if structure perform the following and print appropriate message. i) Get today's date ii) If date is 3, it is dentist appointment. iii) If date is 10, go to conference. iv) If date is other than 3 and 10, no events are scheduled.***

Sol. :

```
<!DOCTYPE html>
<html>
<body>
<?php
 echo "Today date is " . date("d/m/y") . "<br>";
 if((date("d")<3)||(date("d")>10))
       echo "No Event!!!";
 else if((date("d")>3)&&(date("d")<10))
       echo "No Event!!!";
 else
 {
  switch(date("d"))
  {
    case 3 : echo "Dentist Appointment";
          break;
    case 10 : echo "Go to Conference";
          break;
  }
 }
?>
</body>
</html>
```

Output



Ex. 1.5.10 : ***Write a PHP code to display the following pattern***

1
0 1
1 0 1
0 1 0 1
1 0 1 0 1

Sol. :

```
<html>
<head>
</head>
<body>
<?php
 for($i=0;$i<7;$i++)
 {
      for($j=1;$j<$i;$j++)
      {
           if(($i+$j)%2==0)
          {
               printf("0");
          }
               else
               {
                    printf("1");
               }
          }
      print "<br/>";
 }
?>
</body>
</html>
```

Output

❑❑❑

Notes

UNIT - II

2 Arrays, Functions and Graphics

2.1 Introduction to Arrays

- Arrays is a collection of **similar type** of elements, but in PHP you can have the elements of mixed type together in single array.
- In each PHP, each element has two parts **key** and **value**.
- The key represents the index at which the value of the element can be stored.
- The keys are positive integers that are in ascending order.

2.1.1 Creating and Manipulating Arrays

For creating an array in PHP, we use **array()** construct.

Syntax

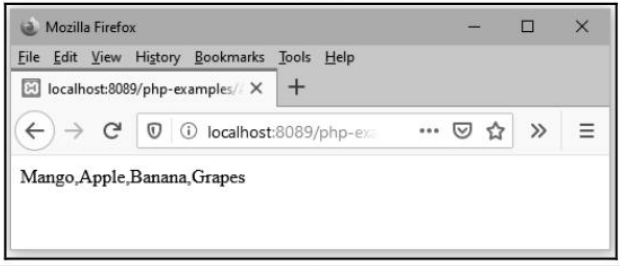
```
$array_name=array(value)
```

Example : Following PHP script shows how to create an array in PHP

ArrayCreateDemo.php

```
<!DOCTYPE html>
<html>
<body>

<?php
$fruits = array("Mango", "Apple", "Banana","Grapes");
echo $fruits[0] . "," . $fruits[1] . "," . $fruits[2] . "," . $fruits[3];
?>
</body>
</html>
```

Output



Program Explanation :

In above program,

(1) We have created an array named **fruits** using **array()** construct

(2) In the echo statement we are displaying each element of the array with the help of array index such as fruits[0],fruits[1],… and so on.

Adding and Deleting Elements

In PHP, **arrays are dynamic.** That means they can grow in size or can shrink.

We can add the element in the array using key/index that hasn't used. For example –

```
$name[5]="CCC";
```

As there is no current value for index 5, the array will grow. Similarly we can skip the index value and add the element as follows –

```
$name[ ]="CCC";
```

The advantage to this approach is that we don't have to worry about skipping an index key.

We can skip some index and can insert the element in the array. For example –

```
$names=array("AAA","BBB","CCC","DDD");
$names[8]="HHH";
```

If we iterate through the array elements then –

```
Array ([0]=>AAA [1]=>BBB [2]=>CCC [3]=>DDD [8] =>HHH)
```

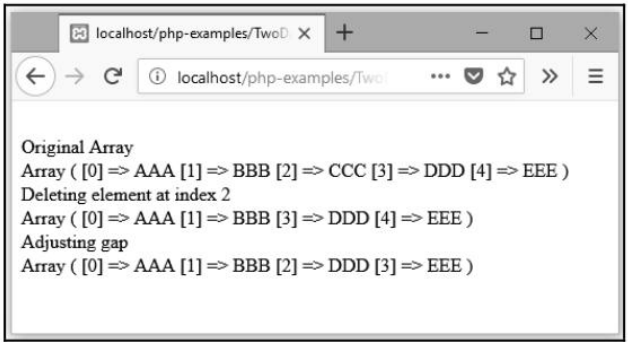
Thus there is now **gap** in our array. If we try to reference the array[4] then it will return NULL value which represents that there is no value present at that index.

Using the **unset** function we can create gaps in the PHP array. For example -

PHP Document

```
<!DOCTYPE html>
<html>
<body>
<?php
$Student =  array("AAA","BBB","CCC","DDD","EEE");
echo "<br/>Original Array<br/>";
print_r($Student);
echo "<br/>Deleting element at index 2<br/>";
unset($Student[2]);
print_r($Student);
echo "<br/>Adjusting gap<br/>";
$Student=array_values($Student);
print_r($Student);
?>
</body>
</html>
```

Output



Script Explanation :

1. In above script we have used **unset** function to create a gap at index 2
2. Later on we have used **array_values()** to adjust this gap by subsequent element.
3. Using the **print_r** function the array can be displayed on the browser window.

Checking if value exists

- The **isset ()** function is used to check whether **a variable is set or not.**
- If a variable is already unset with **unset()** function, it will no longer be set.
- The **isset()** function return false if testing variable contains a NULL value.

PHP Document

```
<!DOCTYPE html>
<html>
<body>
<?php
$Student =  array(1=>"AAA",2=>"BBB");
echo "<br/>Original Array<br/>";
print_r($Student);
if(isset($Student[0]))
 echo "<br/>There is no value set at index 0<br/>";
if(isset($Student[1]))
 echo "The value present at index 1 is: ".$Student[1];
?>
</body>
</html>
```

2.1.2 Types of Arrays

There are three types of arrays

1. Indexed array : Indexed array are the arrays with numeric index. The array values can be stored from index 0. For example –

```
<html>
<head>
    <title>PHP Indexed Arrays</title>
    </head>
   <body>
<?php
$names = array("AAA", "BBB", "CCC");
// Printing array structure
print_r($names);
?>
</body>
</html>
```

Output

Here values gets stored at corresponding index as follows -

```
$mylist[0]  =  10;
$mylist[1]  =  20;
$mylist[2]  =  30;
$mylist[3]  =  40;
$mylist[4]  =  50;
```

We can directly assign some value at specific index.

```
$mylist[5]  =  100;
```

2. Associated array : Associated arrays are the arrays with named keys. It is a kind of array with **name** and **value** pair. For example

```
<html>
<head>
   <title>PHP Associative Array</title>
</head>
<body>
<?php
$city["AAA"] = "Pune";
$city["BBB"] = "Mumbai";
$city["CCC"] = "Chennai";
```

```
// Printing array structure
print_r($city);
?>
</body>
</html>
```

Output

3. Multidimensional Arrays

- PHP support for multidimensional arrays.
- We can store the elements in two dimensional array as

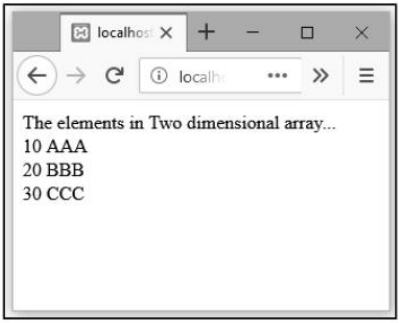
```
$Student = array
(
array(10,"AAA"),
array(20,"BBB"),
array(30,"CCC"),
);
```

- The complete PHP program in which the multidimensional array is created and accessed is as follows-

PHP Document

```
<!DOCTYPE html>
<html>
<body>
<?php
$Student = array
(
 array(10,"AAA"),
 array(20,"BBB"),
 array(30,"CCC"),
 );
echo "The elements in Two dimensional array...<br/>";
for ($row = 0; $row < 3; $row++) {
 for ($col = 0; $col < 2; $col++) {
   echo " ".$Student[$row][$col];
 }
 echo "<br/>";
}
?>
</body>
</html>
```

Output



Review Questions

1. *What is array ? How to create and manipulate arrays ?*
2. *Explain various types of arrays in PHP.*

2.2 Extracting Data from Arrays, Implode, Explode and Flip Functions

(i) The extract function:

Using the extract() function the array keys becomes the variable name and the array values become the variable values.

Syntax

int extract($input_array, $extract_rule, $prefix)

Where

$input_array : is the name of the array whose values need to be extracted. This is a required parameter.

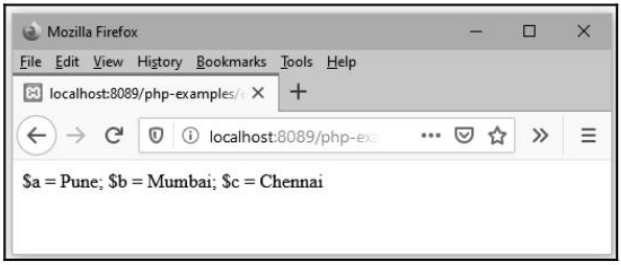
$extract_rule : This is an optional parameter. This parameter specifies how invalid variable names will be treated.

$prefix : The prefix is automatically separated from the array key by an underscore character. This is an optional parameter.

Example Program

```
<!DOCTYPE html>
<html>
<body>

<?php
$arr = array("a" => "Pune","b" => "Mumbai", "c" => "Chennai");
extract($arr);
echo "\$a = $a; \$b = $b; \$c = $c";
?>

</body>
</html>
```

Output



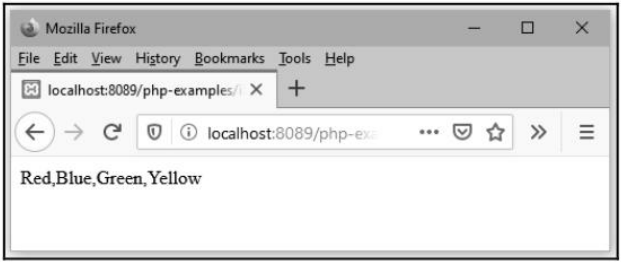
(ii) The implode function :

The implode function converts array into the string. For example

implodeDemo.php

```
<!DOCTYPE html>
<html>
<body>
<?php

        $arr[0] = "Red";
        $arr[1] = "Blue";
        $arr[2] = "Green";
        $arr[3] = "Yellow";
        $text = implode(",",$arr);
        echo $text;

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

Output



(iii) The explode Function

The explode() function is to split a string.

Syntax

```
explode(delimiter, string_name, limit)
```

Where

Delimiter : It sets the boundary string within the input string

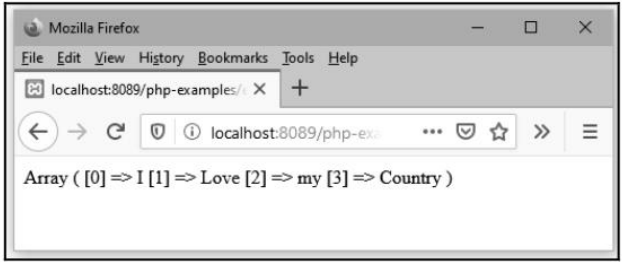
String_name : The name of the string to be split

Limit : It indicates the maximum number of elements in the output array if set to positive value. If set to negative value, all but the last element will be present in the output array.

Example Program

```
<!DOCTYPE html>
<html>
<body>

<?php
$str="I Love my Country";
$arr=explode(" ",$str);
print_r($arr);
?>
</body>
</html>
```

Output



(iv) The flip Function

The array_flip() function is used to exchange the keys with their associated values in array.

That means after applying the array_flip function we get keys from array and those keys become values and values from array become keys.

Syntax

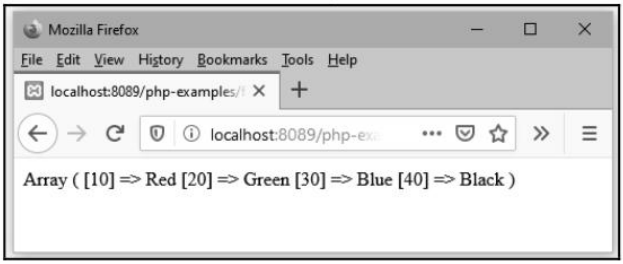
```
array_flip(name_of_array)
```

Example Program

```
<!DOCTYPE html>
<html>
<body>
```

```
<?php
$arr=array("Red"=>10,"Green"=>20,"Blue"=>30,"Black"=>40);
$result=array_flip($arr);
print_r($result);
?>

</body>
</html>
```

Output



Review Question

1. Explain extract, implode, explode and flip function in arrays.

2.3 Traversing Arrays

1. The current and next Function

- The array element reference start at the first element and array maintains an internal pointer using which the next element can be easily accessible.
- This helps to access the array elements in sequential manner.
- The pointer **current** is used to point to the current element in the array. Using the **next** function the next subsequent element can be accessed. Following PHP code illustrates this idea -

PHP Document[ArrayFunDemo4.php]

```
<?php
$mylist = array("Hello", "PHP","You","Are","Wonderful!!!");
$myval=current($mylist);
print("The current value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
```

```
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
?>
```

Output

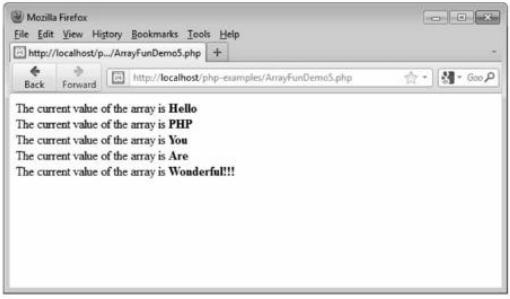
2. The each and foreach Function

- Using **each** function we can iterate through the array elements.

PHP Document[ArrayFunDemo5.php]

```
<?php
$mylist = array("Hello", "PHP","You","Are","Wonderful!!!");
while($myval=each($mylist))
{
 $val=$myval["value"];
 print("The current value of the array is <b>$val</b>");
 print "<br/>";
}
?>
```

Output



- The **foreach** function is used to iterate through all the elements of the loop. The syntax of foreach statement is as follows -

```
foreach($array as $value)
{
  statements to be executed
}
```

- The above code can be modified and written as follows -

PHP Document[ArrayFunDemo6.php]

```
<?php
$mylist = array("Hello", "PHP","You","Are","Wonderful!!!");

foreach($mylist as $value)
{
 print("The current value of the array is <b>$value</b>");
 print "<br/>";
}
?>
```

The output will be the same as above.

Review Question

1. How to traverse an array in PHP ?

2.4 Functions

2.4.1 Defining Function

- The syntax of the function definition is as follows -

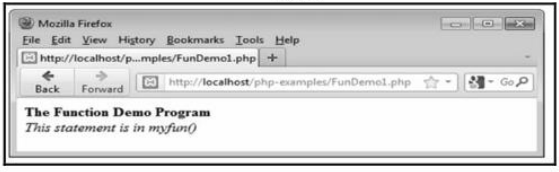
```
function name_of_function(parameter list)
{
  statements to be executed in function
  ...
  ...
  ...}
```

- The function gets executed only after the call to that function. The call to the function can be from anywhere in the PHP code.For example -

PHP Document[FunDemo1.php]

```
<?php
function myfun()
{
  print "<i>This statement is in myfun()</i>";
}
print "<b>The Function Demo Program</b>";
print "<br/>";
myfun();
?>
```

Output

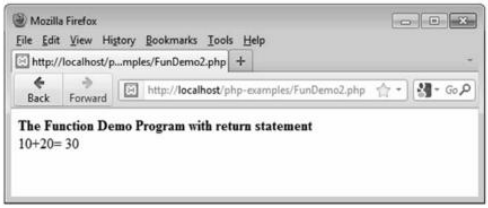


- The **return** statement is used for returning some value from the function body. Following PHP scripts shows this idea.

PHP Document[FunDemo2.php]

```
<?php
function Addition()
{
  $a=10;
  $b=20;
  $c=$a+$b;
  return $c;
}
print "<b>The Function Demo Program with return statement</b>";
print "<br/>";
print "10+20= ".Addition();
?>
```

Output



2.4.2 Parameters

- The parameters that we pass to the function during the call is called the **actual parameter.** These parameters are generally the expressions.
- The parameters that we pass to the function while defining it is called the **formal parameters**. These are generally the variables. It is not necessary that the number of actual parameters should match with the number of formal parameters.
- If there are few actual parameter and more formal parameters then the value of formal parameter is will be some unbounded one.
- If there are many actual parameters and few formal parameters then the excess of actual parameters will be ignored.
- The default parameter passing technique in PHP is **pass by value**. The parameter passing by value means the values of actual parameters will be copied in the formal parameters. But the values of formal parameters will not be copied to the actual parameters.
- Following PHP script illustrates the functions with parameters

PHP Document[FunDemo3.php]

```
<?php
function Addition($a,$b)
{
  $c=$a+$b;
  return $c;
}
print "<b>The Function Demo Program with parameter passing and return statement</b>";
print "<br/>";
$x=10;
$y=20;
print "10+20= ".Addition($x,$y);
?>
```

Output

There are two ways to pass parameters by reference.

1. Add & at the beginning of the name of the formal parameter. For example -

```
<?php
function add_some_extra($string)
{
   $string= 'This string is replaced';
}

$str = 'This is a string ';
$str1=&$str;  //adding & at the beginning of the name of formal parameter.
print "Before function call: $str<br/>";
add_some_extra($str1);
print "After function call: $str<br/>";
?>
```

2. Add & to actual parameter in the function call. For example –

```
<?php
function add_some_extra(&$string)   //adding & to actual parameter in function call
{
   $string= 'This string is replaced';
}
```

```
$str = 'This is a string ';
print "Before function call: $str<br/>";
add_some_extra($str);
print "After function call: $str<br/>";
?>
```

The output of the above code is one and the same it will be as follows -

Output

2.4.3 Types of Functions

1. User defined Functions

User defined functions are the functions created by the user for some specific purpose.

The syntax and example of user defined function is already discussed in section 2.4.1.

2. Variable Functions

- Variable function is a concept in which the variable name is appended with parenthesis.
- In PHP, we can create a variable to which the function is assigned and then that variable is called just like a function.
- Note that the variable function should always be preceded by $ sign.

To understand the concept of variable function let us see one example

Suppose we have three functions namely red(), blue() and green(). These functions can be called with the help of variable function as follows –

VariableFunDemo.php

```
<!DOCTYPE html>
<html>
<body>

<?php
function red()
{
        echo "Roses are Red";
}
function blue()
{
        echo "Sky is Blue";
}
```

```
function green()
{
        echo "Trees are Green";
}
$fun_var="red";
$fun_var();
?>
</body>
</html>
```

You can change it to "blue" or "green"

Calling variable function. $ sign is must

3. Anonymous Function

The anonymous function is a function which is similar to regular function but the **difference** between regular function and anonymous function is that the anonymous function have no name.

Syntax

```
function($argument1,$argument2){
//anonymous function definition
};
```

There are **two important rules** that need to be followed while writing the **anonymous function**

(1) There should not be any function name.

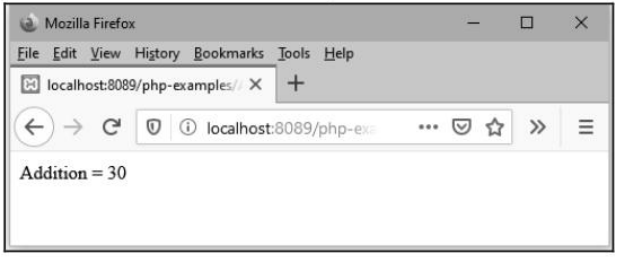
(2) There must be a semicolon after the function definition

Programming Example

AnonymousFunDemo.php

```
<!DOCTYPE html>
<html>
<body>

<?php
$addition=function($arg1,$arg2)
{
 return "Addition = ".($arg1+$arg2);
};
echo $addition(10,20);
?>

</body>
</html>
```

Output



Review Questions

1. *What is function ? How to pass parameters to the function ?*
2. *Explain variable and anonymous function with suitable example in php.*

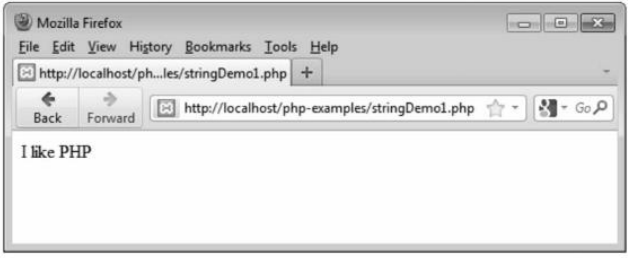
2.5 Strings

- String is a collection of characters.
- In PHP the string is denoted within a double quote.
- Concatenation is the only one operator used in string.It is denoted by dot.
- Strings are treated as the array of characters. The first position of the character is indexed as 0.
- The sample PHP script that stores the string in a variable is as given below -

PHP Script[stringDemo1.php]

```
<?php
$s="I like PHP";
echo $s;
?>
```

Output



Note that the variable **s** is assigned with the string. The string is given in a double quote. Then using the **echo** whatever string is stored in the variable **s** is displayed on the console.

- Various functions used for string handling are -

Function	Purpose	Sample PHP Code	Output
strlen(string1)	It finds the total number of characters in the string	<?php $s="Friend"; echo $s <?php	6
strcmp (string1,string2)	It compares the two strings. It is case sensitive. If this function returns 0 then two strings are equal If this function returns >0 then string1 is greater than string2 If this function returns <0 then string1 is less than string2	<?php echo strcmp('PHP","PHP"); ?>	0
strtolower (string1)	This function converts the characters in string1 to lower case.	<?php echo strtolower('PHP.'); ?>	php
strtoupper (string1)	This function converts the characters in string1 to upper case.	<?php echo strtolower('php'); ?>	PHP
trim(string1)	This function eliminates the white space from both the ends of the string	<?php $str = ' PHP '; echo ' : $str '; echo ' :'.trim($str); echo ' '; ?>	: PHP : PHP

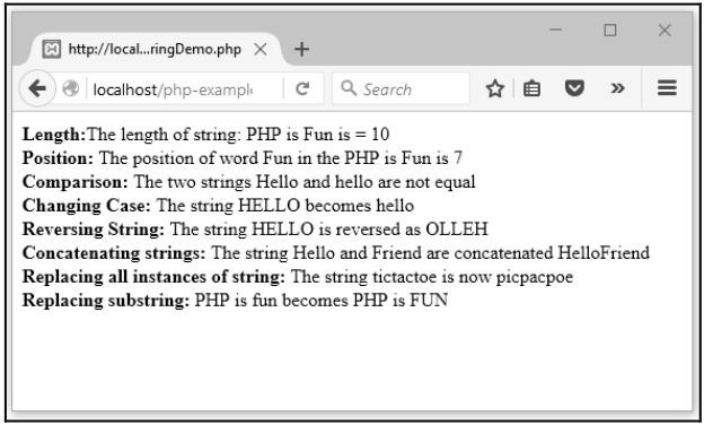
Ex. 2.5.1 : ***Write a PHP program to do string manipulations.***

Sol. : For this program we will apply various built in string manipulating functions to the string. The PHP code is as follows :

```
<?php
$Str1="PHP is Fun";
$length = strlen($Str1);
echo "<b> Length:</b>The length of string: $Str1 is = $length";
echo "<br/><b>Position: </b>The position of word Fun in the $Str1 is ".strpos($Str1,'Fun');
$Str1="Hello";
```

```
$Str2="hello";
if(strcmp($Str1,$Str2))
    echo "<br/><b>Comparison: </b> The two strings $Str1 and $Str2 are not equal";
else
    echo "<br/><b>Comparision: </b> The two strings $Str1 and $Str2 are equal";
$Str1="HELLO";
echo "<br/><b>Changing Case: </b> The string $Str1 becomes ".strtolower($Str1);
echo "<br/><b>Reversing String: </b> The string $Str1 is reversed as ".strrev($Str1);
$Str1="Hello";
$Str2="Friend";
echo "<br/><b>Concatenating strings: </b> The string $Str1 and $Str2 are concatenated ".$Str1.$Str2;
echo "<br/><b>Replacing all instances of string: </b> The string tictactoe is now ";
echo str_replace("t","p","tictactoe");
$Str1="PHP is fun";
$newstring=substr_replace($Str1,"FUN",7,9);
echo "<br><b>Replacing substring: </b> $Str1 becomes $newstring";
?>
```

Output



Review Question

1. What is string ? Explain various operations that can be performed on strings

2.6 Basic Graphics Concepts

- Image is a rectangle of pixels with various colors.
- Any color is composed of three values – Red, Green and Blue ranging from 0 to 255. Here 0 means no-color and 255 means full intensity color. For example – Yellow color is formed with equal values of Red and Green. Here the value of Blue color is 0.
- The image is represented in various file formats such as jpeg, png, giff etc.

- With 256 possible values for each red, blue and green color there are 16,777,216 possible colors for every pixel.

2.6.1 Creating Images

Creating Basic image

The image can be created using following steps

Step 1 : We can create an image in PHP using **imagecreate()** function.

Syntax

```
$image=imagecreate(width, height)
```

Step 2 : The next step is to send a Content-Type header to the browser with the appropriate content type for the kind of image being created. Once that is done, we call the appropriate output function. The ImageJPEG( ) , ImagePNG( ), and ImageWBMP( ) functions create JPEG, PNG, and WBMP files from the image, respectively. This can be done using following code

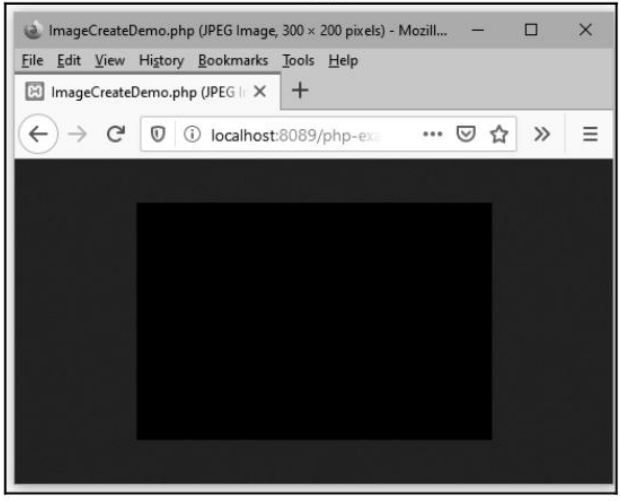
```
imagejpeg($image);
header('Content-Type: image/jpeg');
```

Let us now see a simple program that creates the basic rectangular image

Programming Example

```
<?php
$image = imagecreate(300,200);
imagejpeg($image);
header('Content-Type: image/jpeg');
?>
```

Output



The content-type values are -

Format	Content-Type
GIF	image/gif
JPEG	image/jpeg
PNG	image/png
WBMP	image/vnd.wap.wbmp

Creating an image with some background color

For having an image with some background color we normally use **imagecolorallocate()** function. The syntax for this function is

Syntax

```
Imagecolorallocate(image,color)
```

Where

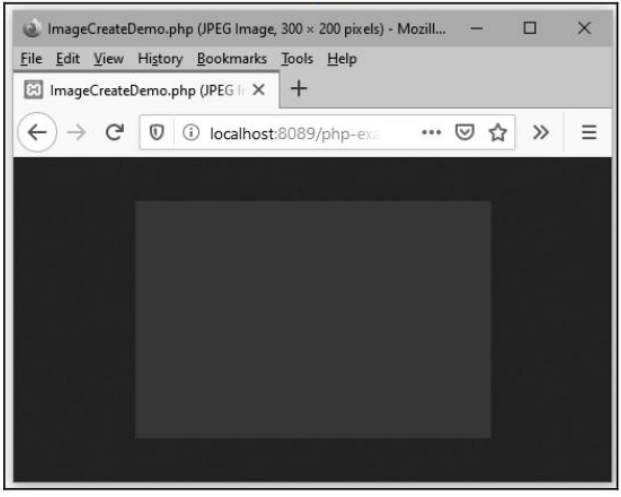
image is the image object

color is given in Red,Green,Blue form. One can specify the color in hexadecimal form as well.

Programming Example

```
<?php
$image = imagecreate(300,200);
$red = imagecolorallocate($image,255,0,0);
imagejpeg($image);
header('Content-Type: image/jpeg');
?>
```

Output



2.6.2 Images with Text

The **ImageString()** method is used to add the text with image. The syntax is

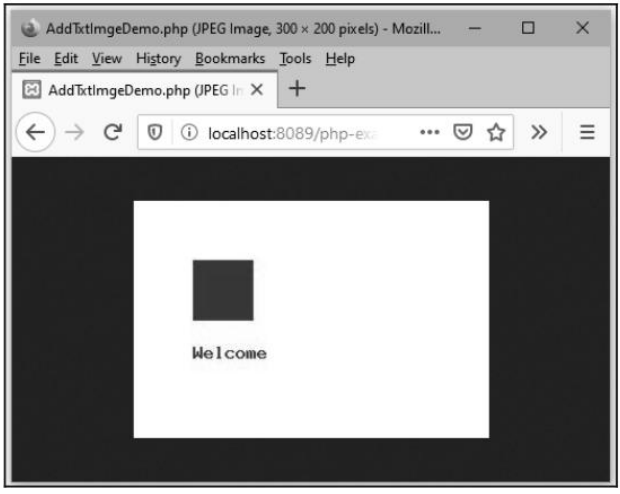
```
ImageString(image, font, x,y,text,color)
```

Programming Example: Following program shows how to add text with the image.

AddTxtImageDemo.php

```
<?php
$image = imagecreate(300,200);
$white = imagecolorallocate($image,255,255,255);
$red = imagecolorallocate($image,255,0,0);
ImageFilledRectangle($image,50,50,100,100,$red);
ImageString($image,5,50,120,"Welcome",$red);
imagejpeg($image);
header('Content-Type: image/jpeg');
?>
```

Output



2.6.3 Scaling Images

Scaling an image means making the image either smaller in size or larger in size than the original.

Using PHP we can resize or scale the image using the function **ImageCopyResampled**

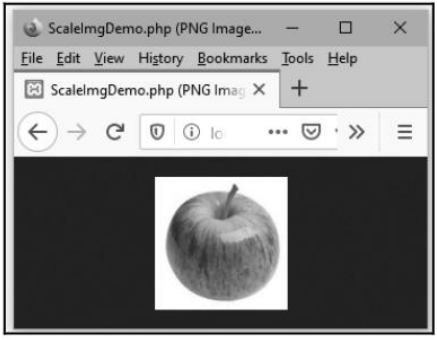
```
imagecopyresampled ( resource $dst_image , resource $src_image ,
                     int $dst_x , int $dst_y ,
                     int $src_x , int $src_y ,
                     int $dst_w , int $dst_h ,
                     int $src_w , int $src_h )
```

Following program illustrates the use of **imagecopyresampled** for scaling the image

ScaleImgDemo.php

```
<?php
$image = imageCreateFromJPEG('d://apple.jpg');← The sample image apple.jpg
$width = ImageSx($image);
$height = ImageSy($image);
$x=$width/2;
$y=$height/2;
$scaledImg=ImageCreateTrueColor($x,$y);
ImageCopyResampled($scaledImg,$image,0,0,0,0,$x,$y,$width,$height);
header('Content-Type: image/png');
ImagePNG($scaledImg);
?>
```

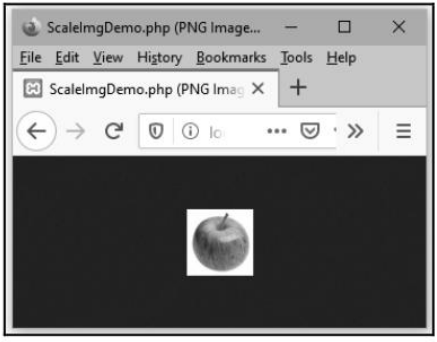
Output



If we change $x and $y value as follows

```
$x=$width/4;
$y=$height/4;
```

Then the output will be



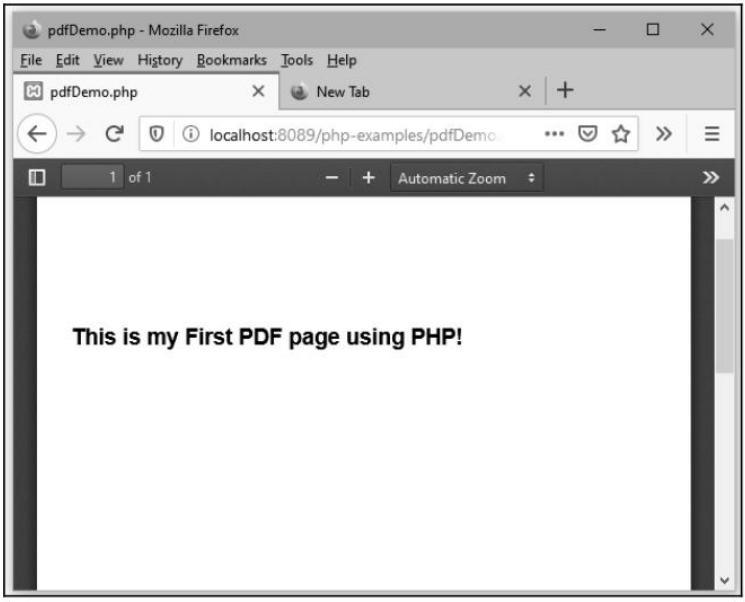
2.6.4 Creation of PDF Document

- For generating a pdf document in PHP, we need some library files.
- FPDF is an open source library which is used for creating a PDF document. It is open source means , we can get it downloaded freely from internet.
- **Features of fpdf:**
  - It is an open source package , hence freely available on internet.
  - It provides the choice of measure unit, page format and margins for pdf page.
  - It provides page header and footer management.
  - It provides automatic page breaks to the pdf document.
  - It provides the support for various fonts, colors, encoding, and image formats.
- If you have downloaded your php using XAMPP package, then **by default** it is present at the location c:/xampp/php/pear/fpdf/
- **Programming Example:** Following is a simple php program that creates a PDF document with some text written on it.

pdfDemo.php

```
<?php
require('c:/xampp/php/pear/fpdf/fpdf.php'); 1
$pdf=new FPDF(); 2
3 $pdf->AddPage();
$pdf->SetFont('Arial','B',20); 4
$pdf->Cell(100,100,'This is my First PDF page using PHP!'); 5
$pdf->Output(); 6
?>
```

Output



Program Explanation : In above program,

(1) We need to include the fpdf library file using **require()** function. The **require()** function is used to put data of one PHP file to another PHP file.

(2) After including the library file, we create an FPDF object. The object is created in variable **$pdf**

(3) There's no page at the moment, so we have to add one with **AddPage()** function.

(4) Before writing any text on the PDF page, we need to set the font. Using **setFont()** function the font is set. The syntax for this function is

```
SetFont(string family , string style , float size)
```

Where

Family : It denotes the font family. It can be 'Arial', 'courier','Times','Symbol and so on.

Style : There are various styles of fond

(i) empty string : regular

(ii) B : bold

(iii) I : italic

(iv) U : underline

Size : Font size is in points.

In above program we are setting 'Arial', bold font with size 20.

(5) A cell is a rectangular area, possibly framed, which contains a line of text. It is output at the current position. We specify its dimensions, its text (centered or aligned), if borders should be drawn.

```
$pdf->Cell(100,100,'This is my First PDF page using PHP!');
```

Using above statement we are writing the text message 'This is my First PDF page using PHP!' at position (x,y)=(100,100).

(6) Finally, the document is closed and sent to the browser with Output().

Review Questions

1. *Write a PHP script to create an image and display text along with the image in PHP.*
2. *Write a PHP program to illustrate scaling of image.*
3. *What are features of fpdf ? Explain how will you create pdf using php.*

❑❑❑

Notes

3 Object Oriented Concepts in PHP

3.1 Creating Classes and Objects

3.1.1 Defining Classes

The syntax for defining classes is class name followed by { }. The properties and methods of class are defined within the braces. For example

```
class Student
{
        public $rollNo;
        public $firstName;
        public $lastName;
        public $age;
}
```

Each property in the class is declared using one of the keywords public, protected, or private followed by the property or variable name.

3.1.2 Instantiating Objects

For making use of the class we must instantiate it. That means **create objects** for the class with the help of **new** operator.

For example – Two objects for the Student class are created in variables $s1 and $s2 as follows –

```
$s1=new Student();
$s2=new Student();
```

3.1.3 Properties

Properties are the data values of the class that can be defined using -> operator. For example

```
$s1=new Student();
$s2=new Student();
$s1->rollNo=101;
$s2->rollNo=102;
```

3.1.4 Method

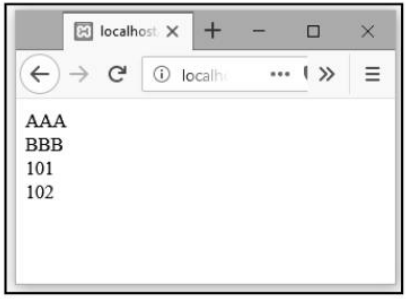
The methods defined in the class specify the functionality of the object. They define the tasks each instance of a class can perform. For example –

PHP Document

```
<?php
    class Student
    {
            /* Member variables */
            var $rollNo;
            var $name;
            /* Member functions */
            function setRollNo($r){
```

```
            $this->rollNo = $r;
        }
        function getRollNo(){
            echo $this->rollNo ."<br/>";
        }
        function setName($nm){
            $this->name = $nm;
        }
        function getName(){
             echo $this->name ." <br/>";
        }
    }

    $Stud1 = new Student();
    $Stud2 = new Student();
    /* Assigning values to properties*/
    $Stud1->setName("AAA");
    $Stud2->setName("BBB");
    $Stud1->setRollNo(101);
    $Stud2->setRollNo(102);

    /* Reading values from properties*/
    $Stud1->getName();
    $Stud2->getName();
    $Stud1->getRollNo();
    $Stud2->getRollNo();
?>
```

Output



3.1.5 Visibility

- Visibility is used to determine the accessibility of a class member.
- There are three modes of visibility and those are - **1. Public, 2. Protected and 3. Private.**
- The **public** keyword means that the property or method is accessible to any code that has a reference to the object. The public properties and method are represented using + in UML class diagram.
- The **private** keyword sets a method or variable to only be accessible from within the class. This means that we cannot access or modify the property from outside of the class. The private properties and method are represented using - in UML class diagram.
- The protected members are used during the inheritance. They are denoted using # in class diagram.

Review Questions

1. *What is class ? Explain it with syntax and example.*
2. *Write a php program to illustrate creation of class and object in PHP.*
3. *Explain properties and methods of OOP in PHP.*
4. *What is visibility in PHP ?*

3.2 Constructors and Destructor

Constructor

- Constructors is a specialized function used to initialize the properties of the class.
- In PHP, the constructor is defined using the function named __construct().
- Each parameter of a class can be assigned using this-> syntax.

- Inside of a class one must always use the $this syntax to reference all properties and methods associated with this particular instance of a class.

For example –

```
class Student
{
...
        function __construct($rollNo,$firstName, $lastName, $age)
        {
         $this->rollNo=rollNo;
         $this->firstName = $firstName;
         $this->lastName = $lastName;
         $this->age = $age;
         }
}
```

The new constructor can be used as follows -

```
$s1=new Student(101,"AAA","BBB",17);
$s2= new Student(102,"XYZ","PQR",25);
```

Destructor

- Destructor is a specialized function which is used to deallocate the memory allocated to it.
- It is denoted as __destruct().
- PHP Destructor method is called just before PHP is about to release any object from its memory. It is normally called before closing the file.
- **Programming Example**

destructDemo.py

```
<!DOCTYPE html>
<html>
<body>

<?php
class Student {
  public $name;

 function __construct($name) {
   $this->name = $name;
 }
 function __destruct() {
   echo "<h4>The Student record with name '{$this->name}' is removed from the database.</h4>";
 }
}

$s = new Student("Ashwini");
?>

</body>
</html>
```

Output

Program Explanation :

In above program,

(1) We have created a class named **Student** with attribute **name**.

(2) The constructor is defined using **__construct** function. When object s is created using following statement.

```
$s = new Student("Ashwini");
```

the constructor function is called by passing **name=Ashwini**.

(3) At the end of PHP program, before closing it, the destructor function is called. The destructor function is defined as **__destruct**.

Review Question

1. *What is constructor and destructor ? How will you use it in PHP ?*

3.3 Inheritance, Overloading and Overriding, Cloning Object

3.3.1 Data Encapsulation

- Data encapsulation is a property in Object oriented programming that refers to restricting access to object's internal components.
- By using data encapsulation, one can **hide** the object's implementation details.
- Generally the **data properties** of the class are made private. Using the public methods of the class these private data members can then be accessed. The **getter and setter** methods are used for that purpose.

Example Code

PHP Document

```
<?php
    class Student
    {
        /* Member variables */
        private $rollNo;
        private $name;
        /* Member functions */
        public function setRollNo($r){
            $this->rollNo = $r;
        }
```

```
        public function getRollNo(){
            echo $this->rollNo ."<br/>";
        }
        public function setName($nm){
            $this->name = $nm;
        }
        function getName(){
            echo $this->name ." <br/>";
        }
    }
    $Stud1 = new Student();
    $Stud2 = new Student();
    /* Assigning values to properties*/
    $Stud1->setName("AAA");
    $Stud2->setName("BBB");
    $Stud1->setRollNo(101);
    $Stud2->setRollNo(102);

    /* Reading values from properties*/
    $Stud1->getName();
    $Stud2->getName();
    $Stud1->getRollNo();
    $Stud2->getRollNo();
?>
```

3.3.2 Inheritance

Inheritance is an important property in object oriented design. This is a property in which one class makes use of some properties of another class.

The class from which another class inherits the properties is called super class or parent class and the class which uses the properties of parent class is called subclass or child class.

PHP makes use of the keyword extends for using inheritance. For instance –

```
class A extends B { . . . }
```

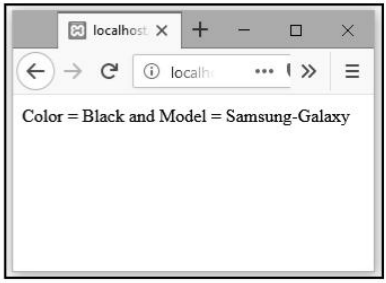
Example Code

PHP Document

```
<?php
//parent class
class Mobile {
    private $color;

    public function setColor($color) {
        $this->color = $color;
    }

    public function getColor() {
        return $this->color;
    }
}

//child class
class Samsung extends Mobile {
    private $model;
```

```
        public function setModel($model) {
                $this->model = $model;
                }

                public function getModel() {
                        return $this->model;
                }
}

//object of the child class
$phone = new Samsung();

//set property
$phone->setColor("Black");
$phone->setModel("Samsung-Galaxy");
//get property
$color = $phone->getColor();
$model = $phone->getModel();

printf("Color = %s and Model = %s", $color, $model);
?>
```

Output



Script Explanation :

In above PHP document -

1) There are two classes created namely – Mobile and Samsung. The Mobile class is a parent class and Samsung class is a child class.
2) The inheritance is achieved using the **extends** keyword.
3) The object named **Phone** is created for the child class **Samsung**.
4) Using the object of child class we can inherit the color property of the parent class.
5) The getter and setter methods are used to set and get the data values of the corresponding classes.

3.3.3 Overloading

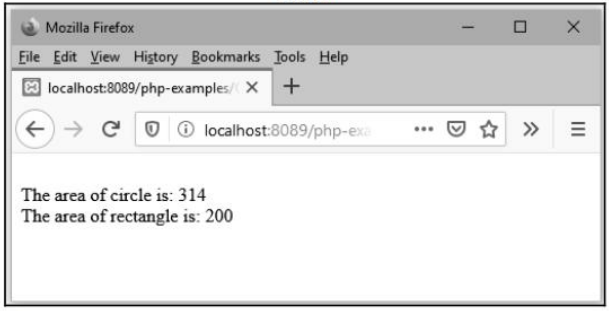
- Function overloading is a mechanism in which there is a same function and the functions perform different tasks.
- Like other object oriented programming languages we cannot use naïve approach.
- In PHP we can perform the method overloading with the help of magic function __call().
- The __call() function takes function name and argument.

OverloadDemo.php

```
<!DOCTYPE html>
<html>
<body>
<?php
class shape {
        public function __call($funname, $arguments)
        {
            if($funname=='area')
            {
                if(count($arguments)==1) {
```

```
                $result=3.14*$arguments[0]*$arguments[0];
                echo "<br/>The area of circle is: ".$result;
                return;
        }
        if(count($arguments)==2) {
                $result=$arguments[0]*$arguments[1];
                echo "<br/>The area of rectangle is: ".$result;
                return;
        }
        else
            echo "Unknown Method";
        }
    }
}
$obj = new shape();
$obj->area(10);
 $obj->area(10,20);
 ?>

 </body>
 </html>
```

Output



Script Explanation : In above PHP program,

(1) We have defined the __call function with the function name **area.**

(2) There can be single argument to the **area** function or there can be two arguments. Depending upon the number of arguments, the function with same name (i.e.area) will either compute area of circle or area of rectangle.

3.3.4 Overriding

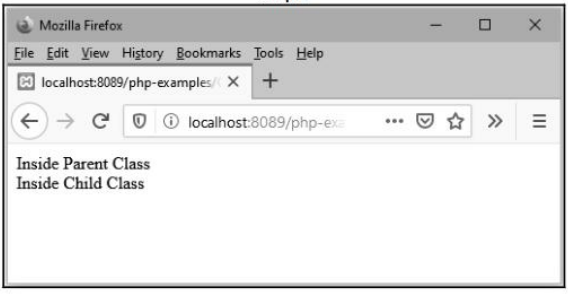
When we inherit a class into another class and provide the definition for one of the function of parent class inside the child class, then this function is overridden and this process is known as function overriding. Note that the name and signature of the function remains the same in the child class as that of parent class, but the function definition is changed.

Following programming example shows the overriding mechanism.

Programming Example

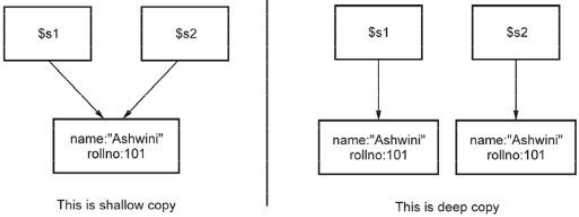
```
<!DOCTYPE html>
<html>
<body>
<?php
class A {
        function fun()
        {
        echo "Inside Parent Class";
        }
}
class B extends A{
        function fun()
       {
          echo "<br/>Inside Child Class";
       }
}
$obj1 = new A();
$obj2 = new B();
$obj1->fun();
$obj2->fun();
?>
</body>
</html>
```

Output



3.3.5 Cloning Object

- **Definition :** Object cloning means creating copy of object.
- We can clone an object in two ways – first one by assignment statement and second one by using __clone operator . This __clone is a magic method.
- If we perform $s2=$s1 then it is called **shallow copy**.
- If we perform $s2= clone $s1 then it is called **deep copy.**
- In shallow copy both the objects point to same reference while in deep copy separate copy of references is created for each object.
- The concept of shallow copy and deep copy is represented by following figure.



Following is an object cloning program that uses clone operator for copying the object.

Programming Example

```
<!DOCTYPE html>
<html>
<body>

<?php
class Student {
        private $name;
        private $rollno;

        function __construct($name,$rollno) {
                $this->name = $name;
                $this->rollno = $rollno;
        }

        function __clone(){
                echo "<h3> Copying object.....</h3>";
        }
        function display()
        {
                echo "<h4>Name: $this->name</h4>";
                echo "<h4>RollNo: $this->rollno</h4>";
        }
}

$s1 = new Student("Ashwini",101);
$s2 = new Student("Sharda",102);
$s1->display();
$s2 = clone $s1;
$s2->display();
?>

</body>
</html>
```

Output

Review Questions

1. *What is inheritance ? How to achieve it in PHP ?*
2. *Explain method overloading and method overriding in PHP.*
3. *What is object cloning? How to clone the object in PHP ?*

3.4 Introspection and Serialization

Introspection is the ability of a program to examine an object's characteristics such as object name, class name, parent class name , method names and so on.

In PHP there some useful functions available for performing introspection. These are –

(1) class_exists() : Checks if the a class has been defined.

(2) get_class() : returns the class name of an object

(3) get_parent_class() : returns the class name of an object's parent class

(4) is_subclass_of() : checks whether an object has a given parent class.

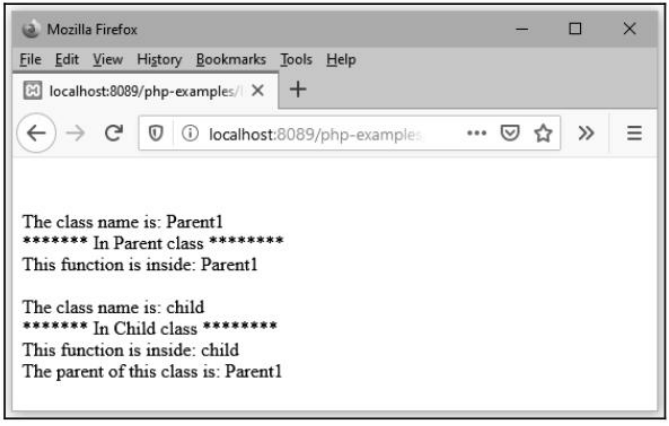
Let us understand the concept of introspection with the help of an example

IntrospectionDemo.php

```
<?php
class Parent1
{
        public function myfun(){
            echo "<br/>******* In Parent class ********";
            echo "<br/>This function is inside: ".get_class($this);
        }
 }
class child extends Parent1
{
        public function myfun() {
```

```
            echo "<br/>******* In Child class ********";
            echo "<br/>This function is inside: ".get_class($this);
            echo "<br/>The parent of this class is: ".get_parent_class($this);
        }
}

if(class_exists("Parent1")){
        $obj_parent1=new Parent1();
        echo "<br/><br/>The class name is: ".get_class($obj_parent1);
        $obj_parent1->myfun();
}
if(class_exists("child")){
        $obj_child=new child();
        echo "<br/><br/>The class name is: ".get_class($obj_child);
        $obj_child->myfun();
}
?>
```

Output



Serialization()

- Serializing an object means **converting it to a byte stream representation** that can be stored in a file.
- This is useful for persistent data. For example, PHP sessions automatically save and restore objects.
- Serialization in PHP requires two functions namely - the **serialize( )** and **unserialize( )**.
- Using **serialize()** method, resulting string is a binary representation of the object and therefore may contain unprintable characters.
- Using **unserialize()** method, the string can be reconstituted back into an object.

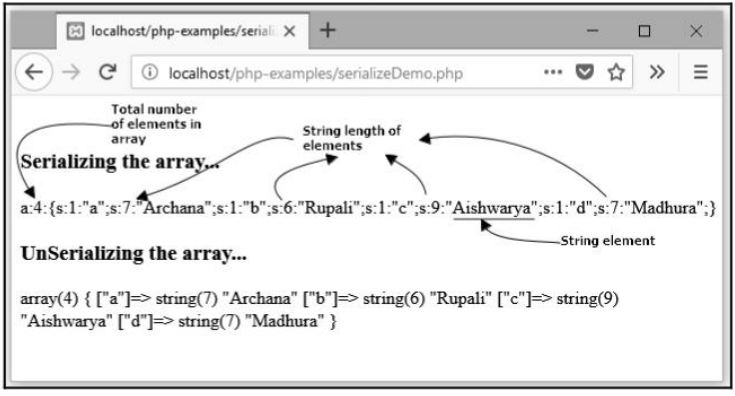
- The **Serializable interface** is used while performing serialization in PHP. It can be represented as follows –

```
interface Serializable {
    public function serialize();
    public function unserialize($serialized);
}
```

- **Programming Example :**

```
<?php
  $array["a"] = "Archana";
  $array["b"] = "Rupali";
  $array["c"] = "Aishwarya";
  $array["d"] = "Madhura";
  print "<br/><h3>Serializing the array...</h3>";
  $str = serialize($array);
  print $str . "\n";
  print "<br/><h3>UnSerializing the array...</h3>";
  $arr = unserialize($str);
  var_dump($arr);
?>
```

Output



Application of Serialization

Various user request of the objects can be stored in serialized form and then at the next request the object is deserialized to **reestablish previous session.** Thus using serialization and unserialization the states of the **requests-responses can be maintained.**

Review Questions

1. *Explain the concept of introspection with suitable PHP program.*
2. *What is serialization ? Explain it with suitable example.*

❑❑❑

UNIT - IV

4 Creating and Validating Forms

4.1 Creating Web Page, GET and POST Methods, Server Role

4.1.1 Creating Web Page

The web page is created using following steps

Step 1 : User creates a web page using form. Various form elements are placed on the form. The user can make a request via web browser with the help of this web page.

Step 2 : The web server fulfills the request and sends back the response to the user in the same web browser.

Step 3 : Finally output is displayed in the browser.

4.1.2 GET and POST Methods

- There are two methods namely – GET and POST for passing the query strings from browser to server.
- Method GET is used to send the query which is less secure. Method POST is used to send more secured data.

Difference between GET and POST Methods

Sr. No.	GET Request	POST Request
1.	**Parameters** remain in **browser history** because they are part of the URL.	**Parameters** are **not saved in browser history.**
2.	GET is **less secure** compared to POST because data sent is part of the URL. So it is saved in browser history and server logs in plaintext.	POST is a little **safer than GET** because the parameters are not stored in browser history or in web server logs.
3.	This request can be cached.	This requests are never cached.
4.	This request can be **bookmarked.**	This request **can not be bookmarked.**
5.	GET method **should not be** used when sending passwords or other **sensitive information.**	POST method **used when sending passwords** or other sensitive information.
6.	Only **limited amount of information is sent** using GET request.	**Large amount of information** is sent using POST request.
7.	It **is more efficient.**	It **is less efficient.**

Working of GET and POST

- If we have a Input form on which there are some text boxes placed. The user fills up the information within the text boxes and clicks the submit button. The information within the text boxes will then be transmitted to the server via **query string**. This scenario can be represented by following figure -

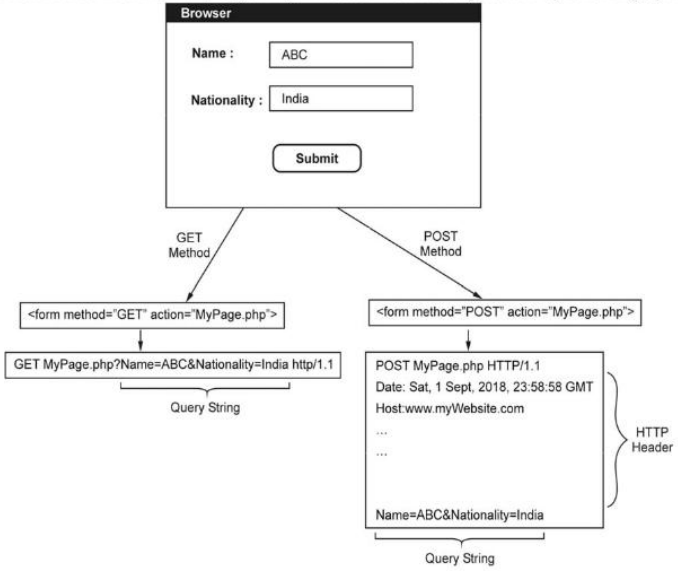
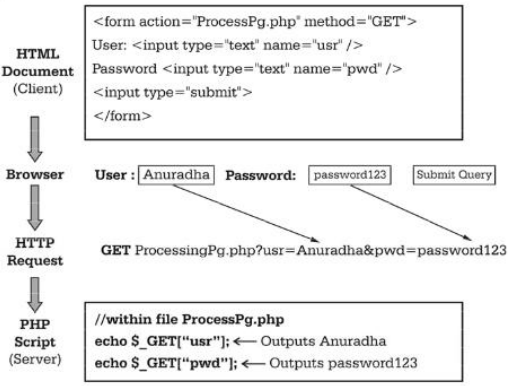


Fig. 4.1.1 Information processing by GET and POST methods

Using Query string the information is transmitted to the server.

4.1.3 Concept of Subglobal Array

- PHP is used for form handling. For that purpose the simple form can be designed in XHTML and the values of the fields defined on the form can be transmitted to the PHP script using GET and POST methods.
- For forms that are submitted via 'GET' method, we can obtain the form via the **$_GET** array variable.
- For forms that are submitted via 'POST' method, we can obtain the form via the **$_POST** array variable.
- The $_GET and $_POST arrays are the most important superglobal variables in PHP since they allow the programmer to access data sent by the client in a query string.
- It works as follows -



4.1.4 Server Role

The responsibilities of web server can be illustrated by enlisting following functionalities :

(1) The primary responsibility of web server is to **respond to the requests** made by the Web clients.

(2) It is responsible to **HTTP connections.**

(3) Web server manages **permissions and access** for certain resources.

(4) It is responsible for **encrypting and compressing** data.

(5) Web server **manages multiple domains** and URLs

(6) It is **handles database connections, cookies,** states and **uploading of files.**

4.1.4.1 Apache and PHP

- PHP is a **server-side** language that can be **directly installed** as an Apache module. Refer Fig. 4.1.2.

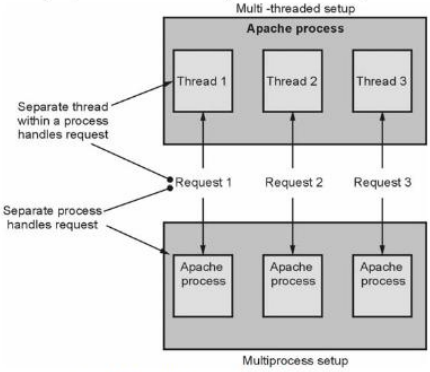


Fig. 4.1.2 Multi-threaded vs multi-process

- The PHP module **mod_php5** is sometimes referred as Server Application Programming Interface**(SAPI)**. This layer handles the interaction between the PHP environment and web server environment.
- Apache **works in two modes -**
  - Multi-process or preforked
  - Multi-threaded or worker
- In **Multi-process mode -**
  - Basically the default installation of Apache is in multi-process mode in which each request is handled using **separate process.**
  - A **fork** is an activity in which operating system creates a copy of already running process.
  - Forking is efficient in UNIX system but is slow in Windows OS.
  - The main advantage of multi-process mode is that is process is separated from other process, hence one process can't affect other process.
- **In multi-threaded mode -**
  - Each process runs multiple threads.
  - A thread is basically a lightweight process intended to perform some task and it runs within an operating system process. Refer Fig. 4.1.2.
  - A thread uses less memory than a process and shares the memory of the belonging process.
  - As consequence, the multi-threaded mode typically works better with large load.

4.1.4.2 Installation of Apache, PHP and MySQL

For installing Apache, PHP and MySQL an all in package like XAMPP/WAMPP is preferred.

Ex. 4.1.1 : ***Explain how can you create a web based application using XAMPP. Give all the steps required in detail.***

Sol. : XAMPP is a free distribution package that makes it easy to install Apache Web Server, MySQL, PHP, PEAR. Here in XAMPP(The X stands for any OS) or WAMPP(the W stands for Windows OS).

Step 1 : Go to the site **https://www.apachefriends.org/index.html**

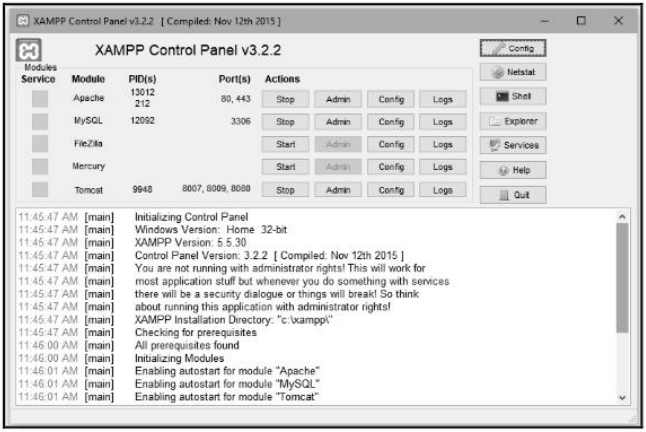
Step 2 : Click on Download XAMPP for Windows or Linux depending upon your operating system.

Step 3 : When prompted for the download, click 'Save' and wait for your download to finish.

Step 4 : Install the program, and click on 'Run.' Accept default settings by clicking Next button. Finally you will get installation completion message.

Step 5 : On your drive, the XAMPP folder will be created. Click on xampp_start file, this will enable to start Apache, MySQL and Tomcat start.

Step 6 : The control panel for XAMPP will look like this



Step 7 : Write a PHP script and save it in **C:\XAMPP\htdocs\php-examples** folder by giving the filename and extension as .php

Step 8 : Open the web browser and type **http://localhost/php-examples/yourfilename.php**

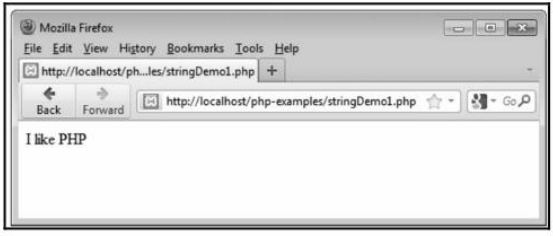
Step 9 : The web application will be executed within your web browser.

For example

PHP Script[stringDemo1.php]

```
<?php
$s="I like PHP";
echo $s;
?>
```

Output



Review Questions

1. *Explain the method of passing information via query strings.*
2. *What is $_GET and $_POST superglobal arrays ? Explain its working.*
3. *Write a short note on - Apache and PHP.*
4. *Explain PHP internals in brief.*

4.2 Form Controls

- The HTML form is defined using the <form> element.
- Typical component of forms are **text, text area, checkboxes, radio buttons and push buttons.**
- HTML allows us to place these form components on the web page and send the desired information to the destination server.
- All these form contents appear in the <form> tag.
- The form has an attribute action which gets executed when user clicks a button on the form.
- Various attributes of form are –

Attribute	Description
action	It specifies the url where the form should be submitted.
method	It specifies the HTTP methods such as GET, POST
name	This attribute denotes the name of the form.
target	It specifies the target of the address in the action attribute.

Let us learn various form components with the help of simple HTML documents.

4.2.1 Text Box

- **Text** is typically required to place one line text. For example if you want to enter some name then it is always preferred to have Text field on the form.
- The text field can be set using

```
<input type="text" size="30" name ="username" value=" ">
```

- The input type is **text** and the **value** of this text field is " " That means the blank text field is displayed initially and we can enter the text of our choice into it. There is **size** parameter which allows us to enter some size of the text field.
- Some other parameters or attributes can be
  - **maxlength** that allows us to enter the text of some maximum length.
  - **name** indicates name of the text field.
  - **align** denotes the alignment of the text in the text field. The alignment can be left, right, bottom and top.
- **Programming Example**

Step 1 : Create a form in some HTML file as follows –

Input.html

```
<form action="hello.php" method="get">
      Name: <input type="text" name="user"/>
      <input type="submit" value="Submit"/>
</form>
```

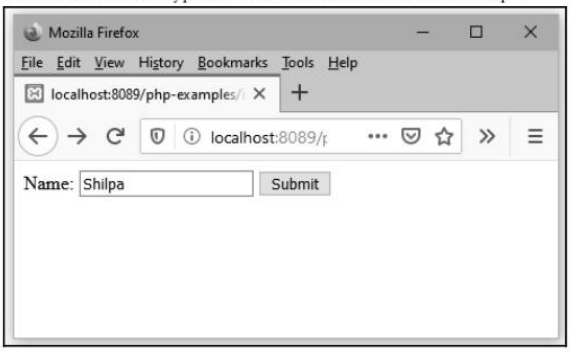
Step 2 : Now the **hello.php** program can be as follows -

Hello.php

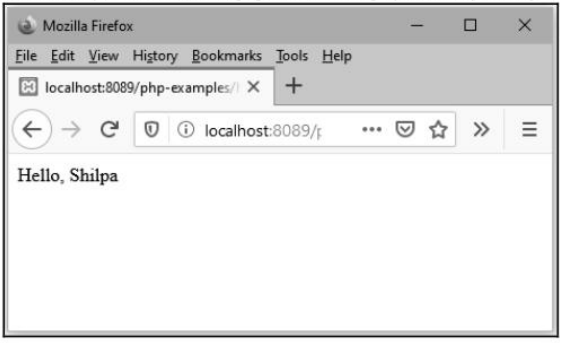
```
<?php
$name=$_GET["user"];
```

```
echo "Hello, $name";
?>
```

Step 3 : Open the web browser and type the name of the HTML file created in step 1



Click Submit button, and you can invoke the php file. It will display following message



In above code, we can use **$_POST** instead of **$_GET** array.And when we use $_POST array in the PHP script, the form method on HTML form must **post**. That means the form code in HTML form must be

```
<form action="hello.php" method="post">
```

4.2.2 Text Area

Text field is a form component which allows us to enter single line text, what if we want to have multiple line text ,then you must use textarea component.

Syntax

```
<textarea name=name_of_component rows="some_number" cols="some_number"></textarea>
```

Various parameters that can be set for the text area can be

rows denotes total number of rows in the text area.

cols specifies total number of columns in the text area.

name denotes the name of the text area which can be utilized for handling that component for some specific purpose.

Following example illustrate how to create an application that uses the textarea component in PHP.

Step 1 :

Input.html

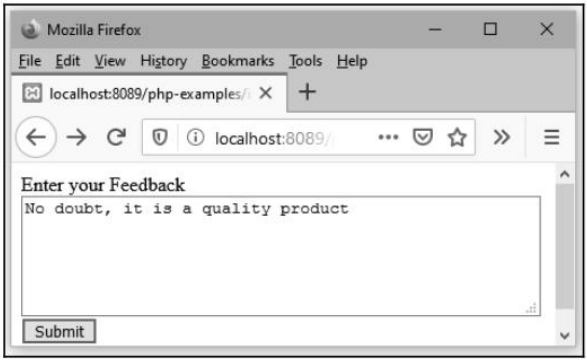
```
<form action="Info.php" method="get">
        Enter your Feedback<br/>
        <textarea name="feedback" rows="5" cols="50"></textarea>
        <br/>
        <input type="submit" value="Submit"/>
</form>
```

Step 2 :

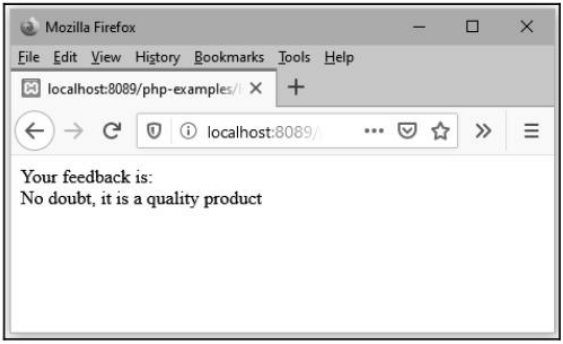
Info.php

```
<?php
$data=$_GET["feedback"];
echo "Your feedback is: <br/>".$data;
?>
```

Output



Click Submit button, and PHP script will be invoked



4.2.3 Radio Button

- This form component is also use to indicate the selection from several choices.
- Using input type="radio" we can place radio button on the web page.
- **Syntax**

  <input type="radio" name="name" value="Value_in_number">
- This component allows us to make only one selection at a time.
- We can create a group of some radio button component.
- **Programming Example**

Step 1 :

Input.html

```
<form action="Info.php" method="get">
        Please Select your favorite fruit:<br/>
        <input type="radio" name="fruit" value="Mango">Mango<br/>
        <input type="radio" name="fruit" value="Banana">Banana<br/>
        <input type="radio" name="fruit" value="Grapes">Grapes<br/>
        <br/>
        <input type="submit" value="Submit"/>
</form>
```

Step 2 :

Info.php

```
<?php
$choice=$_GET["fruit"];
if(($choice!=null))
{
        echo("My favorite fruit is: ".$choice);
}
?>
```

Output

Click Submit button and the php page will be displayed as follows –

4.2.4 Check Box

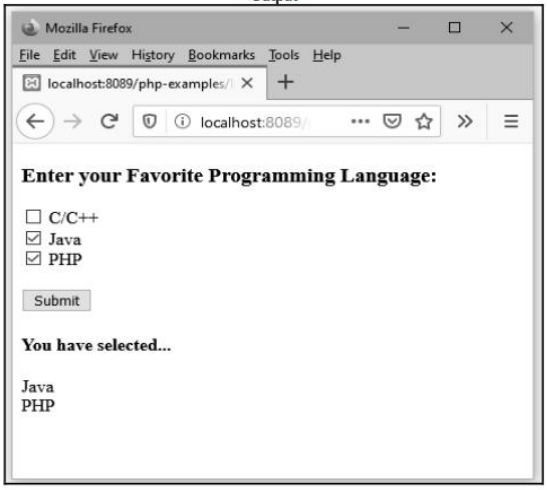
- It is the simplest component which is used particularly when we want to make some selection from several options.
- For having the checkbox we have to specify the input type as checkbox.
- **Syntax**
  <input type="checkbox" name="name_of_option" value="value_in_number" checked="checked">Mango
- If we want to get the checkbox displayed as checked then set checked='checked'
- **Programming Example**

chkDemo.php

```
<form action="#" method="post">
<h3> Enter your Favorite Programming Language:</h3>
```

```
<input type="checkbox" name="check_list[]" value="C/C++">
<label>C/C++</label><br/>
<input type="checkbox" name="check_list[]" value="Java">
<label>Java</label><br/>
<input type="checkbox" name="check_list[]" value="PHP">
<label>PHP</label>
<br/><br/>

<input type="submit" name="submit" value="Submit"/>
<br/>
</form>
<?php
if(isset($_POST['submit']))//to run PHP script on submit
{
        echo "<h4>You have selected... </h4>";
        if(!empty($_POST['check_list']))
        {
            // Loop to store and display values of individual checked checkbox.
                foreach($_POST['check_list'] as $selected)
                {
                        echo $selected."</br>";
                }
        }
}
?>
```

Output



4.2.5 List

PHP allows us to have **List** on the web page so that the desired selection can be made.

The parameter **select** is for the List component and **option** parameter is for setting the values to the options of the list.

Syntax

```
<select name=" ... ">
        <option value=" ">label</option>
        <option value=" ">label</option>
        <option value=" ">label</option>
        ...
</select>
```

Following example shows how to make use of list in PHP

ListDemo.php

```
<form action="#" method="post">
<h3> Select your Favorite Programming Language:</h3>
<h4>(Press ctrl+click for multiple selection)</h4>
<select name="Language[]" multiple>//initializing name with array
        <option value="C/C++">C/C++</option>
        <option value="C#">C#</option>
        <option value="Java">Java</option>
        <option value="PHP">PHP</option>
        <option value="Python">Python</option>
</select>
<br/><br/>
<input type="submit" name="submit" value="Submit"/>
<br/>
</form>
<?php
if(isset($_POST['submit']))//to run PHP script on submit
{
        echo "<h4>You have selected... </h4>";
        foreach($_POST['Language'] as $selected)
        {
                echo $selected."</br>";
        }
}
?>
```

Output

4.2.6 Button

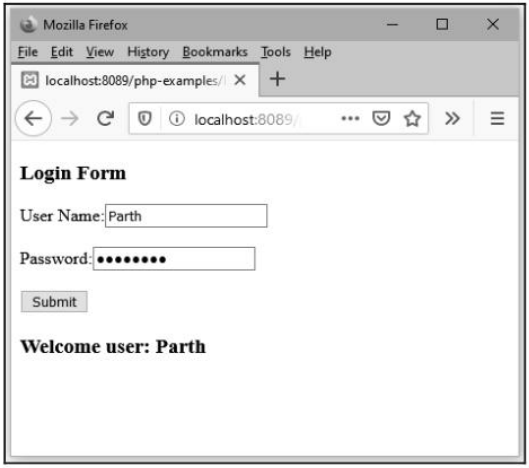
We can create the button using <input type = 'submit'>

Various parameters of submit button are

1) name denotes the name of the submit button.
2) value is for writing some text on the text on the button.

ButtonDemo.php

```
<form action="#" method="post">
<h3> Login Form</h3>
User Name:<input type="text" name="username"/>
<br/><br/>
Password:<input type="password" name="password"/>
<br/><br/>
<input type="submit" name="submit" value="Submit"/>
<br/>
</form>
<?php
if(isset($_POST['submit']))//to run PHP script on submit
{
        echo "<h3> Welcome user: ".$_POST['username']."</h3>";
}
?>
```



Examples Based on Form Controls

Ex. 4.2.1 : *Create HTML form with one textbox to get user's name. Also write PHP code to show length of entered name when, the HTML form is submitted.*

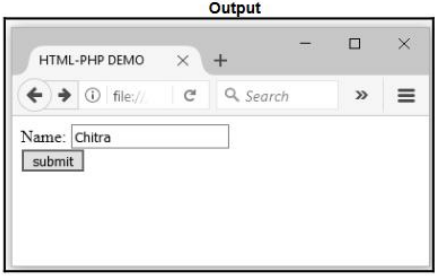
Sol. : **Step 1 :** The HTML form can be created as follows

```
<!DOCTYPE html>
<html>
<head><title> HTML-PHP DEMO</title>
</head>
<body>
   <form method="post" action="http://localhost/getdata.php">
        Name: <input type="text" name="myname" size="20"/>
        <br/>
        <input type="submit" name="submit" value="submit"/>
   </form>

</body>
</html>
```

Step 2 : The PHP script to display the length of submitted name is as written below

```
<?php
print "The name is: ";
print $_POST["myname"];
$len= strlen($_POST["myname"]);
print "<br/> The length of name is: ";
print $len;
?>
```

Output



Ex. 4.2.2 : *Create HTML form to enter one number. Write PHP code to display the message about number is odd or even.*

Sol. :

Step 1 : The HTML form for accepting number is created as below -

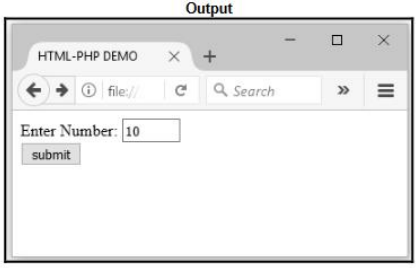
```
<!DOCTYPE html>
<html >
 <head><title> HTML-PHP DEMO</title>
</head>
<body>
   <form method="post" action="http://localhost/getdata.php">
        Enter Number: <input type="text" name="mynum" size="5"/>
        <br/>
        <input type="submit" name="submit" value="submit"/>
   </form>
</body>
</html>
```

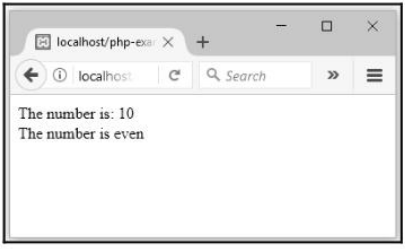
Step 2 : The PHP script deciding whether the number is even or odd is as given below -

```
<?php
print "The number is: ";
print $_POST["mynum"];
$a= $_POST["mynum"];
if($a%2==1)
 print "<br/> The number is odd ";
```

```
else
 print "<br/> The number is even ";
?>
```

Output





Ex. 4.2.3 : ***Create a form containing information Sr.no, title of the book, publishers, quantity, price read the data from the form and display it using PHP script.***

Sol. : Step 1 : Create an HTML page for inputting the data. Following is the code for HTML script .

HTML Document[input.html]

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
<title> Book Order Form </title>
</head>
<body>
<h3> Enter the Book Data </h3>
<form method="post" action="http://localhost/php-examples/formdemo.php">
<table>
<tr>
<td>Sr.No.</td>
<td><input type="text" name="SLNo"></td>
</tr>
<tr>
 <td>Book name</td>
 <td><input type="text" name="BName"></td>
```

```
</tr>
<tr>
 <td>Publisher</td>
 <td><input type="text" name="PUBName"></td>
</tr>
<tr>
 <td>Price</td>
 <td><input type="text" name="Price"></td>
</tr>
 <td>Quantity</td>
 <td><input type="text" name="Qty"></td>
</tr>
<tr>
 <td><input type="submit" value="Submit"></td>
 <td><input type="submit" value="Clear"></td>
</tr>
</table>
</form>
</body>
</html>
```

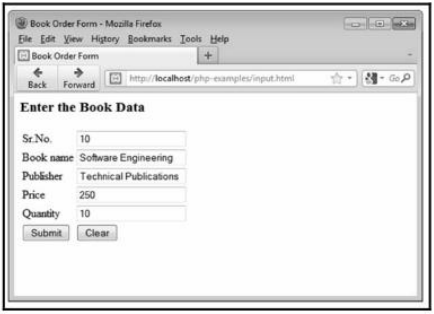
Step 2 : Create a PHP script which will read out the data entered by the user using HTML form. The code is as follows -

PHP Document[formdemo.php]

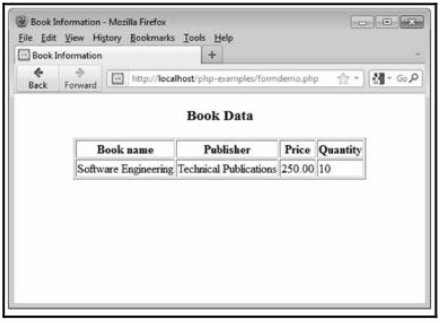
```
<html>
<head>
 <title>Book Information</title>
</head>
<body>
<?php
$BName=$_POST["BName"];
$PUBName=$_POST["PUBName"];
$Price=$_POST["Price"];
$Qty=$_POST["Qty"];
?>
<center>
<h3> Book Data </h3>
<table border=1>
<tr>
<th>Book name</th>
 <th>Publisher</th>
 <th>Price</th>
 <th>Quantity</th>
</tr>
<tr>
 <td><?php print ("$BName"); ?></td>
 <td><?php print ("$PUBName"); ?></td>
 <td><?php printf("%3.2f",$Price); ?></td>
```

```
<td><?php printf("%d",$Qty); ?></td>
</tr>
</table>
</center>
</body>
</html>
```

Step 3 : Open some suitable web browser and enter the address for the HTML file which you have created in step 1.



Now click on the Submit button and the PHP file will be invoked. The output will then be as follows -



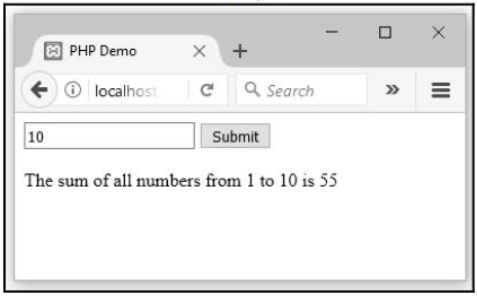
Ex. 4.2.4 : ***Write a PHP program to accept a positive integer 'N' through a HTML form and to display the sum of all the numbers from 1 to N.***

Sol. :

```
<html>
    <head>
        <title> PHP Demo </title>
    </head>
    <body>
        <form method="post">
```

```
                    <input type="text" name="num"/>
                    <input type="submit" value="Submit"/>
                </form>
        </body>
</html>
<?php
        //getting values from HTML form
        $n = intval($_POST['num']);
        $sum=0;
        for($i=1;$i<=$n;$i++)
                $sum=$sum+$i;
        echo "The sum of all numbers from 1 to ".$n." is ".$sum;

?>
```

Output



Review Question

1. *Explain date and time control in HTML5 with necessary illustrations.*

4.3 Working with Multiple Forms

4.3.1 A Web Page Having Many Forms

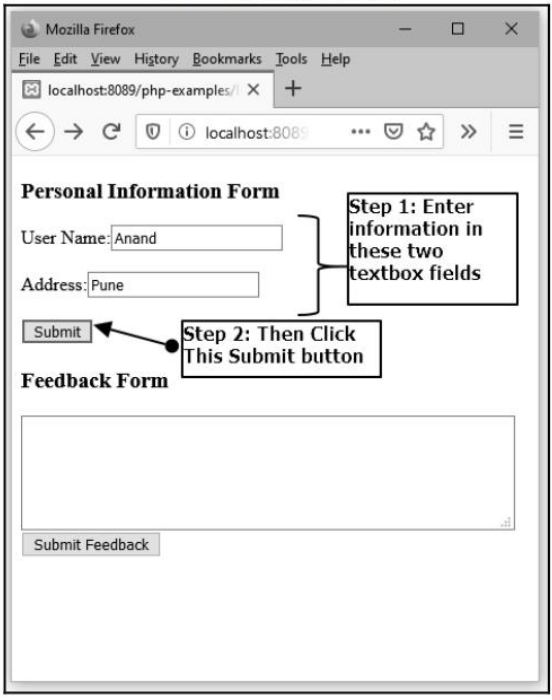
In some web applications, we can have multiple forms in a single PHP document. Following program shows how to handle the requests from multiple forms.

ManyFormDemo.php

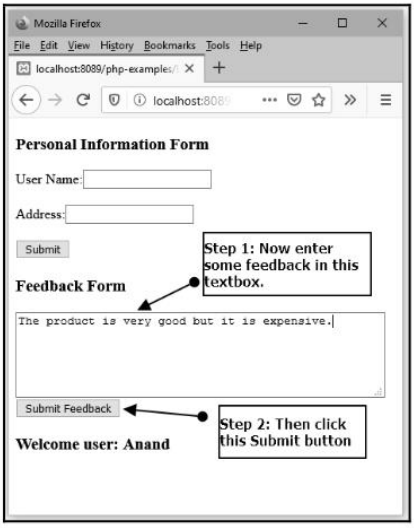
```
<form method="post">
<h3> Personal Information Form</h3>
User Name:<input type="text" name="username"/>
<br/><br/>
Address:<input type="text" name="address"/>
<br/><br/>
<input type="submit" name="submit_personal_info" value="Submit"/>
<br/>
</form>
<form method="post">
```

```
<h3> Feedback Form</h3>
<textarea name="feedback" rows="5" cols="50"></textarea>
<br/>
<input type="submit" name="submit_feedback" value="Submit Feedback"/>
</form>

<?php
if(!empty($_POST['submit_personal_info']))
{
        echo "<h3> Welcome user: ".$_POST['username']."</h3>";
}
if(!empty($_POST['submit_feedback']))
{
        echo "<h3> We value your feedback: </h3>";
        echo "Your feedback is<br/>".$_POST["feedback"];
}
?>
```

Output
(on Submitting Personal_info form)



(on Submitting feedback form)


Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost:8089/php-examples/
localhost:8089
Personal Information Form
User Name:
Address:
Submit
Feedback Form
The product is very good but it is expensive.
Submit Feedback
Welcome user: Anand
Step 1: Now enter some feedback in this textbox.
Step 2: Then click this Submit button


Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost:8089/php-examples/
localhost:8089
Personal Information Form
User Name:
Address:
Submit
Feedback Form
Submit Feedback
We value your feedback:
Your feedback is
The product is very good but it is expensive.

4.3.2 A Form Having Multiple Submit Buttons

One can create a PHP page with multiple submit buttons on single form. The action indicated by each submit button is separately handled in PHP.

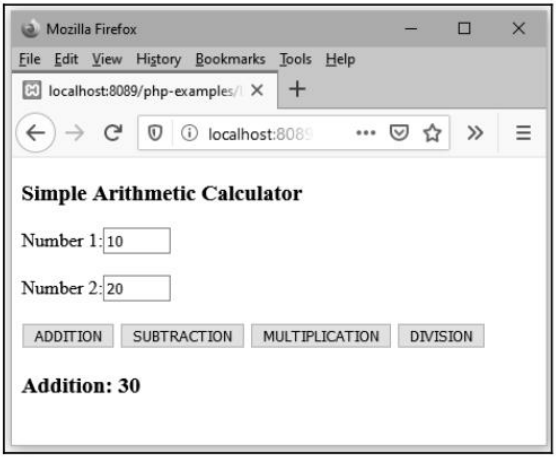
Following is a calculator program, in which basic arithmetic operations are triggered by four submit buttons.

Calculator.php

```
<form method="post">
<h3> Simple Arithmetic Calculator</h3>
Number 1:<input type="text" size="5" name="num1"/>
<br/><br/>
Number 2:<input type="text" size="5" name="num2"/>
<br/><br/>
<input type="submit" name="add" value="ADDITION"/>
<input type="submit" name="sub" value="SUBTRACTION"/>
<input type="submit" name="mul" value="MULTIPLICATION"/>
<input type="submit" name="div" value="DIVISION"/>

<?php
if(!empty($_POST['add']))
{
        $result=$_POST['num1']+$_POST['num2'];
        echo "<h3> Addition: ".$result."</h3>";
}
if(!empty($_POST['sub']))
{
        $result=$_POST['num1']-$_POST['num2'];
        echo "<h3> Subtraction: ".$result."</h3>";
}
if(!empty($_POST['mul']))
{
        $result=$_POST['num1']*$_POST['num2'];
        echo "<h3> Multiplication: ".$result."</h3>";
}
if(!empty($_POST['div']))
{
        $result=$_POST['num1']-$_POST['num2'];
        echo "<h3> Division: ".$result."</h3>";
}

?>
```

Four Submit buttons on a single form

Output



4.4 Web Page Validation

preg_match function : The preg_match() function searches string for pattern, returning true if pattern exists, and false otherwise.

Syntax

int preg_match (string $pattern , string $subject [, array &$matches [, int $flags = 0 [, int $offset = 0]]])

where

pattern denotes the pattern to be searched for

subject denotes the input string

$matches[0] will contain the text that matched the full pattern

The optional parameter offset can be used to specify the alternate place from which to start the search.

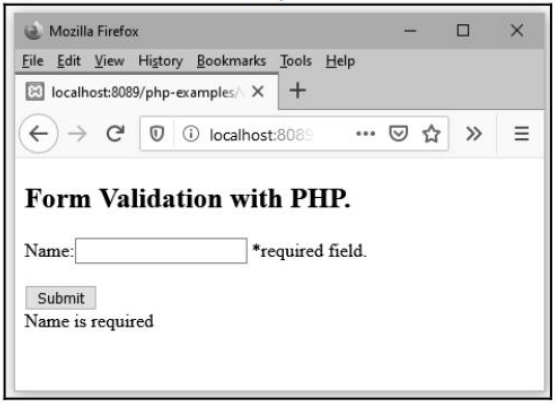
Checking for name field

Validation.php

```
<!DOCTYPE html>
<html>
<body>
    <h2>Form Validation with PHP.</h2>
    <form method="post">
        Name:<input name="name" type="text" value="">
        <span> *required field.</span>
        <br/><br/>
        <input name="submit" type="submit" value="Submit">
```

```
        </form>
</body>
</html>
<?php
// On submitting form below function will execute.
if(isset($_POST['submit'])){
        if (empty($_POST["name"]))
        {
                echo "Name is required";
        }
        else
        {
                $name = test_input($_POST["name"]);
        // check name only contains letters and whitespace
                if (!preg_match("/^[a-zA-Z ]*$/",$name))
                {
                        echo "Only letters and white space allowed";
                }
        }
}
function test_input($data)
{
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
}
//php code ends here
?>
```

Output

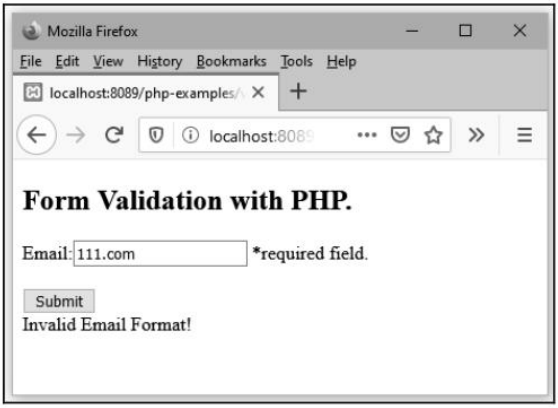


Checking Email Field

validation.php

```
<!DOCTYPE html>
<html>
<body>
        <h2>Form Validation with PHP.</h2>
        <form method="post">
                Email:<input name="email" type="text" value="">
                <span> *required field.</span>
                <br/><br/>
                <input name="submit" type="submit" value="Submit">
        </form>
</body>
</html>
<?php

// On submitting form below function will execute.
if(isset($_POST['submit'])){
          if (empty($_POST["email"]))
          {
               echo "Email is required";
          }
          else
          {
               $email = test_input($_POST["email"]);
        // check if e-mail address syntax is valid or not
               if (!preg_match("/([w-]+@[w-]+.[w-]+)/",$email))
               {
                     echo "Invalid Email Format!";
               }
          }
}
function test_input($data)
{
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
}
//php code ends here
?>
```

Output



4.5 Cookies

- Cookie is a small file that server embeds in the user's machine. This is another method of passing user information to the server.
- Cookies are used to identify the users.
- A cookie consists of **a name** and a **textual value**. A cookie is created by some software system on the server.
- In every HTTP communication between browser and server a **header** is included. The header stores the information about the message.
- The header part of http contains the cookies.
- There can be one or more cookies in browser and server communication.

Uses of Cookies

- While cookies can be used for any state-related purpose, they are principally used as a way of **maintaining continuity** over time in a web application.
- One typical use of cookies in a website is to **"remember" the visitor**, so that the server can customize the site for the user.
- Some sites will use cookies as part of their **shopping cart implementation** so that items added to the cart will remain there even if the user leaves the site and then comes back later.
- Cookies are also frequently used to **keep track** of whether a user has **logged into a site**.

4.5.1 Working of Cookies

The working of cookies can be illustrated by following steps –

Step 1 : User makes first request to page on web site www.website.com

Step 2 : Page sets cookie values as part of response.

Step 3 : HTTP response contains cookies in header.

Step 4 : Browser saves the cookies in a text file and associate this cookie file with www.website.com.

Step 5 : User makes another request to the page on the site www.website.com

Step 6 : Browser reads cookie values from text file for each subsequent request for www.website.com.

Step 7 : Cookie values travel in every subsequent HTTP request for that domain.

Step 8 : Server for www.website.com retrieves these cookie values from request header and uses them to customize the response.

- There are two types of cookies : **session cookies** and **persistent cookies.**
- A **session cookie** has no expiry stated and thus will be deleted at the end of the user browsing session.
- **Persistent cookies** have an expiry date specified; they will persist in the browser's cookie file until the expiry date occurs, after which they are deleted.

4.5.2 Using Cookies

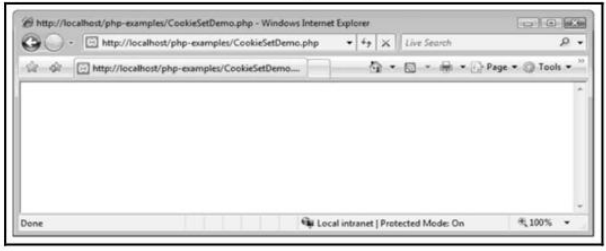
- PHP can be used to create and retrieve the cookies.
- The cookie can be set in PHP using the function called **setcookie()**
- The syntax for the cookie is -

  setcookie(name,value,expire_period,path,domain)
- Following is a simple PHP document which illustrates how to set the cookies -

PHP Document[CookieSetDemo.php]

```
<?php
$Cookie_period=time()+60*60*24*30;
setcookie("Myname", "Monika", $Cookie_period);
?>
```

Output



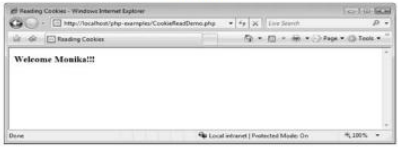
Note that you have got the blank screen it indicates that the cookie is set. In above PHP document we have set the PHP script for one month. Just observer the third parameter of the setcookie function.

Now you can retrieve the cookie and read the value to ensure whether or not the cookie is set.

PHP Document[CookieReadDemo.php]

```
<html>
 <head><title>Reading Cookies</title>
<body>
 <?php
if (isset($_COOKIE["Myname"]))
  echo "<h3>Welcome " . $_COOKIE["Myname"]."!!!</h3>";
else
  echo "<h3>Welcome guest!</h3>";
 ?>
</body>
</html>
```

Output



Program Explanation

- The **isset** function is used for checking whether or not the cookie is set.
- Then using the $_COOKIE the value of the cookie can be retrieved.

Review Questions

1. *Explain the concept of cookies with its working.*
2. *Write a PHP script to demonstrate creation and reading of cookies.*

4.6 Session Handling

- When you open some application, use it for some time and then close it. This entire scenario is named as **session.**
- **Session state** is a server-based state mechanism that lets web applications store and retrieve objects of any type for each **unique user session.** That is, each browser session has its own session state stored as a serialized file on the server, which is deserialized and loaded into memory as needed for each request. Refer Fig. 4.6.1.

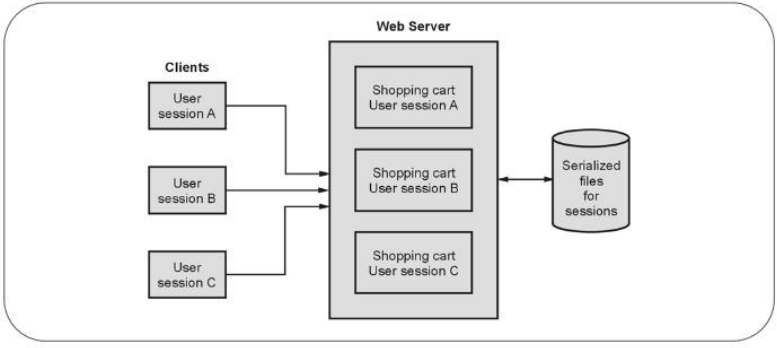
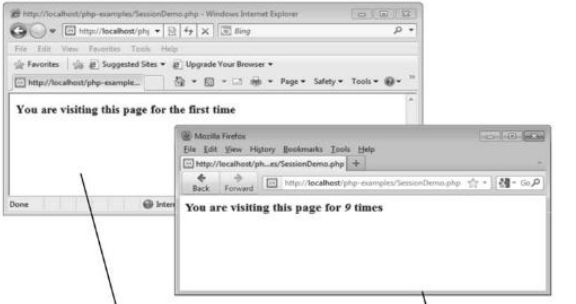


Fig. 4.6.1 Session state

- Because server storage is a finite resource, objects loaded into memory are released when the request completes, making room for other requests and their session objects. This means there can be more active sessions on disk than in memory at any one time.
- Sometimes the information about the session is required by the server. This information can be collected during the session. This process is called **session tracking.**
- In PHP, session state is available to the developer using the $_SESSION variable. The unique ID for the sessions can be stored in superglobal array $_SESSION.
- PHP keeps track of session by using a function called **session_start().** Due to the call to **session_start()** function the session ID is created and recorded.
- Following is a simple PHP script in which the information about session is tracked.

PHP Document[SessionDemo.php]

```
<?php
session_start();
if(isset($_SESSION['pgvisit']))
{
 $_SESSION['pgvisit']=$_SESSION['pgvisit']+1;
echo "<h3>You are visiting this page for <i>". $_SESSION['pgvisit']."</i> times</h3>";
}
else
{
 $_SESSION['pgvisit']=1;
echo "<h3>You are visiting this page for the first time</h3>";
}
?>
```



Open some another web browser say Internet Explorer and type the URL for **SessionDemo.php**. For the first time the above message will be displayed

Open the Mozilla Firefox web browser and type the URL for **SessionDemo.php**. Just refresh this page for many times.

Program Explanation :

- In above program, we have started the session by using **session_start()** function.
- The **isset()** function checks if the **pgvisit** variable has already been set. If **pgvisit** has been set, we can increment our counter. If **pgvisit** is not set, then we create a set it to 1.
- The value of **pgvisit** is displayed on the screen.

Review Question

1. *Explain the working of session handling in PHP.*

4.7 Sending Email

PHP has a **mail()** function which is useful in sending the mail from the script. The syntax of this function is -

mail(to,subject,message,headers,parameters);

to : represents the address of receiver.

Subject : It specifies the subject of mail.

Message : It defines the message which is to be sent.

Header : This is optional and specifies the additional headers like Cc, Bcc.

Parameters : This is optional and specifies the additional parameters.

Following is a simple PHP script that illustrates how to use **mail** function

```
<html>
<head>
<title> Mail() function Demo in PHP</title>
</head>
<body>
<?php
$to = "abc.aaa@gmail.com";
$subject = "Test Mail";
$txt = "Hello, how are you?";
$headers = "From: xyz_xxx@gmail.com" . "\r\n" .
"CC: mnp_pqr@gmail.com";
$staus=mail($to,$subject,$txt,$headers);
if($status== true)
{
        echo "Message is sent!!!";
}
else
{
        echo "Error:Message can not be sent!!!";
}
?>
</body>
</html>
```

Review Question

1. *Explain the use of mail() function in PHP.*

❑❑❑

Notes

UNIT - V

5 Database Operations

5.1 Introduction to MySQL

- MYSQL is a open source database product and can be downloaded from the web site http://dev.mysql.com/downloads/mysql.
- MYSQL is a kind of database in which the records are stored in an entity called **tables.**
- In the tables the data is arranged in the **rows and columns**.
- We can query a database to retrieve particular information. **Query** is a request or a question for the database. There is a common practice of making use of structured query language(SQL).

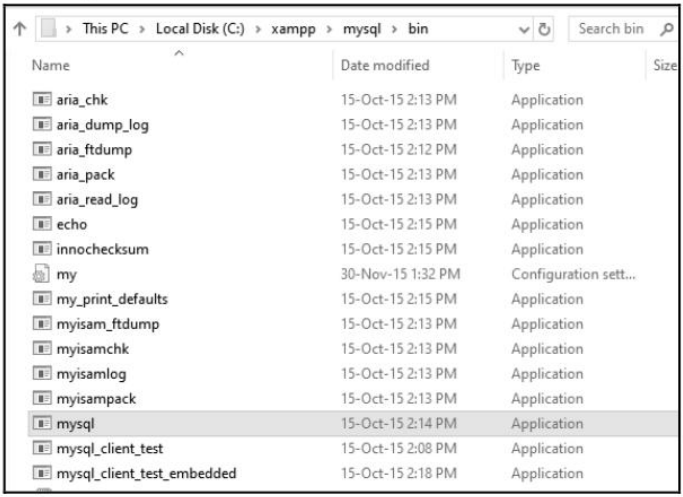
Advantages of PHP-MySQL

- PHP is a server side scripting language and it has an ability to create dynamic pages with customized features. Using PHP-MySQL user friendly and interactive web sites can be created.
- Both PHP and MySQL are open-source technologies that work hand-in-hand to create rich internet applications. The purchased code provides you the encrypted source code to prevent replication or modification, whereas open-source programs encourage users to utilize, scrutinize and customize the code.
- Due to availability of these technologies as free of cost, the cost effective web solutions can be created.
- PHP-MySQL are stable technologies and have cross platform compatibility. Hence the web application developed using these technologies become portable.
- Since HTML can be embedded within the PHP, there is no need to write separate code for web-scripting.
- Open-source coding has been checked and doubled checked by thousands or even millions of people around the world. Hence one can built the reliable web application using these technologies.
- The most popular web sites being developed using PHP and MySQL technologies are -
  1. Facebook 2. WordPress
  3. Wikipedia 4. Yahoo

5.1.1 Handling MySQL Queries

After installing MySQL or XAMPP, we can use MySql prompt for submitting the SQL queries. With the help of SQL queries we can create database, insert data to database, update or delete data from the database.

We can get the command prompt for MySQL by executing **mysql.exe** file present in MySQL folder. As I have installed XAMPP, I could locate MySQL folder as



Just double click the **mysql.exe** file and you will get the command prompt **mysql>**

Following are the illustrations that help us to perform various operations on database using MySQL queries.

1. Creating database

```
mysql> CREATE DATABASE mydb;
Query OK, 1 row affected (0.15 sec)
```

2. Displaying all the databases

```
mysql> SHOW DATABASES;
+-------------+
| Database |
+-------------+
| mydb       |
| mysql      |
| students   |
| test       |
+--------------+
4 rows in set (0.06 sec)
```

3. Selecting particular database

```
mysql> USE MYDB;
Database changed
```

4. Creating table

We must create a table inside a database hence it is a common practice to use create table command after USE database command. While creating a table we must specify the table fields.

```
mysql> CREATE TABLE my_table(id INT(4),name VARCHAR(20));
Query OK, 0 rows affected (0.28 sec)
```

5. Displaying a table

After creating the table using SHOW command we can see all the existing tables in the current database.

```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_mydb |
+----------------------+
| my_table |
+----------------------+
1 row in set (0.00 sec)
```

6. Displaying the table fields

For knowing the various fields of the table we may use following command

```
mysql> DESCRIBE my_table;
+---------+---------------+--------+------+---------- +---------+
| Field | Type | Null | Key | Default | Extra |
+---------+---------------+--------+------+----------+--------+
| id | int(4) | YES | | NULL | |
| name | varchar(20) | YES | | NULL | |
+---------+---------------+--------+------+---------- +-------- +
2 rows in set (0.07 sec)
```

7. Inserting values into the table

We can insert only one complete record at a time. It is as shown below –

```
mysql> INSERT INTO my_table
   -> VALUES(1,'SHILPA');
Query OK, 1 row affected (0.05 sec)
```

8. Displaying the contents of the table

```
mysql> SELECT * FROM my_table;
+------+------------+
| id | name |
+------+------------+
| 1 | SHILPA |
+-------+-----------+
1 row in set (0.06 sec)
```

We can also write SELECT statement for selecting particular row by specifying some condition such as -

```
mysql> SELECT * FROM my_table where id=1;
```

or

```
mysql> SELECT * FROM my_table where name='SHILPA';
```

Thus we can insert the rows into the table by repeatedly giving the INSERT command.

If we want to get the records in sorted manner then we use ORDER BY clause

```
mysql> SELECT * FROM my_table;
+-------+--------------+
| id    | name         |
+-------+--------------+
|   1   | SHILPA       |
|   2   | SUPRIYA      |
|   3   | YOGESH       |
|   4   | MONIKA       |
+-------+--------------+
4 rows in set (0.00 sec)
mysql> SELECT * FROM my_table ORDER BY name;
+------+-------------+
| id   | name        |
+------+-------------+
|   4  | MONIKA      |
|   1  | SHILPA      |
|   2  | SUPRIYA     |
|   3  | YOGESH      |
+------+-------------+
4 rows in set (0.00 sec)
```

9. Updating the record

For updating the record in the database following command can be used -

```
mysql> UPDATE my_table
   -> SET name='PRIYANKA'
   -> WHERE id=4;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> SELECT * FROM my_table;
+------+--------------+
| id   | name         |
+-----+--------------+
|   1 | SHILPA       |
|   2 | SUPRIYA      |
|   3 | YOGESH       |
|   4 | PRIYANKA     |
+-----+--------------+
4 rows in set (0.00 sec)
```

10. Deleting record

For deleting particular record from a database

```
mysql> DELETE FROM my_table
   -> WHERE id=3;
Query OK, 1 row affected (0.04 sec)
```

Then use SELECT statement for displaying the contents of the table we use following command

```
mysql> SELECT * FROM my_table;
+------+---------------+
| id   | name          |
+------+---------------+
|    1 | SHILPA        |
|    2 | SUPRIYA       |
|    4 | PRIYANKA      |
+------+---------------+
3 rows in set (0.00 sec)
```

11. For deleting the table

The table can be deleted using the command

```
mysql>drop table my_table;
```

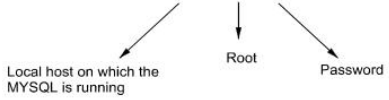
Review Questions

1. Explain the benefits of PHP and MySQL.

5.2 Connecting to MySQL Database

- MYSQL is a kind of database in which the records are stored in an entity called **tables.**
- In the tables the data is arranged in the rows and columns. We can query a database to retrieve particular information. **Query** is a request or a question for the database. There is a common practice of making use of structured query language (SQL).
- The PHP function **mysql_connect** connects to the MYSQL server. There are three parameters that can be passed to this function. For example –

```
mysql_connect("localhost", "root", "mypassword") or die(mysql_error());
```



- The database can be selected by using the command **mysql_select_db.**

  For example

```
mysql_select_db("test")
```

will select the database named **test.**

1. Creation of Database

- We can create a database using the function **mysql_query**. The **mysql_error()** function is used to obtain the error messages if any command gets failed.
- **mysql_query** function in php is used to pass a sql query to mysql database.

Syntax

```
mysql_query ( string query [, resource link_identifier])
```

This function returns the query handle for SELECT queries, TRUE/FALSE for other queries, or FALSE on failure.

Example

```
mysql_query("CREATE DATABASE mydb",$conn)
```

- The **mysql_connect()** function Open a connection to a MySQL Server.

Syntax

```
mysql_connect ( string server , string username , string password)
```

Returns a MySQL link identifier on success, or FALSE on failure.

Example

```
$conn=mysql_connect("localhost ","root","mypassword");
```

- The **mysql_close()** function is used to close the database connection.

Syntax

```
mysql_close (Connection)
```

PHP Example For Creating Database

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost ","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
//Create a database
if(mysql_query("CREATE DATABASE mydb",$conn))
{
 print "Database created";
}
else
{
 print "Error creating database : " . mysql_error();
}
mysql_close($conn); //closing the database
?>
```

2. Selecting Database

The database can be selected using the function **mysql_select_db()**.

Syntax

```
mysql_select_db ( string database_name [, resource link_identifier])
```

Where

mysql_select_db() attempts to select existing database on the server associated with the specified link identifier. It returns TRUE on success, or FALSE on failure.

For example -

```
<?php
// Make a MySQL Connection
```

```
$conn=mysql_connect("localhost :3306/mydb ","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
//Select a database
mysql_select_db("mydb", $conn);
mysql_close($conn); //closing the database
?>
```

3. Creation of Table

Before creating the table a database must be created and within which the table can be created. Note that before creating a table the desired database must be selected.

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost ","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
//Create a database
if(mysql_query("CREATE DATABASE mydb",$conn))
{
print "Database created";
}
else
{
  print "Error creating database : " . mysql_error();
}
mysql_select_db("mydb",$conn); //before creating table select the database
$query="CREATE TABLE my_table
        (
          id INT(4),
          name VARCHAR(20)
        )";
mysql_query($query,$conn);//Execution of Query

mysql_close($conn); //closing the database
?>
```

5.3 Insertion of Data, Retrieve Query Result

1. Insertion of Data

For inserting a data into the table we use the INSERT query. For example

```
$query=" INSERT INTO my_table (id,name)  VALUES(1,'SHILPA')";
mysql_query($query,$conn);//Execution of Query
```

Here is a PHP script in which insert query is used to insert two records in the table

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn); //select the database
$query=" INSERT INTO my_table (id,name)  VALUES(1,'SHILPA')";
mysql_query($query,$conn);//Execution of Query
$query=" INSERT INTO my_table (id,name)  VALUES(2,'MONIKA')";
mysql_query($query,$conn);//Execution of Query

mysql_close($conn); //closing the database
?>
```

Sometimes values that can be inserted in the table can be obtained from some another script and these values might be present in the variables. Insertion of such data can be done using $_POST variables. It is as shown below -

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn); //select the database
$query=" INSERT INTO my_table (id,name)
VALUES('$_POST[MyID]','$_POST[MyName]')";
mysql_query($query,$conn);//Execution of Query

mysql_close($conn); //closing the database
?>
```

2. Retrieving the Query Result

For displaying the records present in the database table, we use SELECT query. For example

```
//Execution of Query for displaying the data
$result=mysql_query("SELECT * FROM my_table");
```

The above execution returns a result handle. Then The **mysql_fetch_array()** is used to retrieve a row of data as an array from a MySQL result handle.

Purpose of mysql_fetch_array() : The mysql_fetch_array() is used to retrieve a row of data as an array from a MySQL result handle.

Syntax :

```
mysql_fetch_array(result, result_type)
```

PHP Script for Displaying records

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost ","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn); //select the database
//Execution of Query for displaying the data
$result=mysql_query("SELECT * FROM my_table");
 while($row = mysql_fetch_array($result))
  {
   echo $row['id'] . " " . $row['name'];                    //Each record will be displayed
                                                            //line by line
   echo "<br />";
  }
mysql_close($conn); //closing the database
?>
```

3. Counting Number of Rows in Table

The number of rows present in the database table can be obtained using **mysql_num_rows** function.

Syntax

int mysql_num_rows (resource $result)

This returns number of rows in result on success, or NULL on error.

Example

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost :3306/mydb ","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
//Select a database
mysql_select_db("mydb", $conn);
$num_rows = mysql_num_rows($result);
//Print number of rows
echo "Total number of rows are $num_rows";
mysql_close($conn); //closing the database
?>
```

4. Counting Number of Fields in Table

The **mysql_num_fields()** is used to get number of fields of the table.

Syntax

mysql_num_fields(resource_name)

It returns the number of fields present in the resource and false on failure

Example

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost:3306/mydb ","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
//Select a database
mysql_select_db("mydb", $conn);
$result = mysql_query("select id,name from my_table where id = '1' ");
echo mysql_num_fields($result); // since two fields are fetched, returns 2
?>
```

Review Questions

1. *Explain with PHP code, how to connect to a database ?*
2. *Write a PHP code to retrieve data from database.*

5.4 Update and Delete Operations on Table Data

1. Updating Data From Table

We can update data from the database using UPDATE query.

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn); //select the database
$query="UPDATE mytable SET phone='55555' WHERE phone='22222'";
mysql_query($query,$conn);//Execution of Query
mysql_close($conn); //closing the database
?>
```

2. Deleting Data From Table

We can delete the values from the database using the DELETE query

For example

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
```

```
mysql_select_db("mydb",$conn); //select the database
$query=" DELETE FROM my_table WHERE id=1";
mysql_query($query,$conn);//Execution of Query
mysql_close($conn); //closing the database
?>
```

Similarly we can delete a database using the query DROP.

For example –

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost","root","mypassword");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn); //select the database
$query=" DROP DATABASE mydb";
mysql_query($query,$conn);//Execution of Query
mysql_close($conn); //closing the database
?>
```

Review Questions

1. Write a PHP program to update the database.

2. How will you delete record from database using PHP program ?

5.5 Programming Examples using PHP-MYSQL

Ex 5.5.1 : ***Write a PHP script to create a new database table with 4 fields of your choice and perform following database operations. i) Insert ii) Update iii) Delete***

Sol. : We will create a table in the database **test**. The name of the table is **mytable**. Then we will insert the record into the table using the INSERT command, update particular field of the record using the command UPDATE and Delete the record using the command DELETE.

The PHP script is as fo llows -

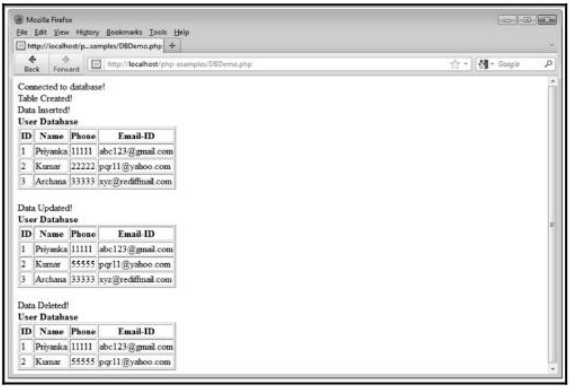
PHP Document[DBDemo.php]

```
<?php
// Make a MySQL Connection
mysql_connect("localhost", "root", "mypassword") or die(mysql_error());
mysql_select_db("test") or die(mysql_error());
echo "Connected to database!";
mysql_query("CREATE TABLE mytable(id INT NOT NULL AUTO_INCREMENT, PRIMARY KEY(id), name VARCHAR(30),
 phone INT,emailId VARCHAR(30))")
 or die(mysql_error());
print "<br/>";
echo "Table Created!";
```

```
// Insert a row of information into the table "example"
mysql_query("INSERT INTO mytable
(name, phone,emailId) VALUES('Priyanka', '11111','abc123@gmail.com' ) ")
or die(mysql_error());

mysql_query("INSERT INTO mytable
(name, phone,emailId) VALUES('Kumar', '22222','pqr11@yahoo.com' ) ")
or die(mysql_error());

mysql_query("INSERT INTO mytable
(name, phone,emailId) VALUES('Archana', '33333','xyz@rediffmail.com' ) ")
or die(mysql_error());
print "<br/>";
echo "Data Inserted!";
$result =mysql_query("SELECT * FROM mytable")
or die(mysql_error());
print "<br/>";
print "<b>User Database</b>";
echo "<table border='1'>";
echo "<tr><th>ID</th> <th>Name</th> <th>Phone</th><th>Email-ID</th> </tr>";
while($row = mysql_fetch_array( $result ))
 {
 // Print out the contents of each row into a table
 echo "<tr><td>";
 echo $row['id'];
 echo "</td><td>";
 echo $row['name'];
 echo "</td><td>";
 echo $row['phone'];
 echo "</td><td>";
 echo $row['emailId'];
 echo "</td></tr>";
}
echo "</table>";
$result = mysql_query("UPDATE mytable SET phone='55555' WHERE phone='22222'")
or die(mysql_error());

print "<br/>";
echo "Data Updated!";
$result =mysql_query("SELECT * FROM mytable")
or die(mysql_error());
print "<br/>";
print "<b>User Database</b>";
echo "<table border='1'>";
echo "<tr><th>ID</th> <th>Name</th> <th>Phone</th><th>Email-ID</th> </tr>";
while($row = mysql_fetch_array( $result ))
 {
 // Print out the contents of each row into a table
```

```
 echo "<tr><td>";
 echo $row['id'];
 echo "</td><td>";
 echo $row['name'];
 echo "</td><td>";
 echo $row['phone'];
 echo "</td><td>";
 echo $row['emailId'];
 echo "</td></tr>";
}
echo "</table>";
$result = mysql_query("DELETE FROM mytable WHERE phone='33333'")
or die(mysql_error());
print "<br/>";
echo "Data Deleted!";
$result =mysql_query("SELECT * FROM mytable")
or die(mysql_error());
print "<br/>";
print "<b>User Database</b>";
echo "<table border='1'>";
echo "<tr><th>ID</th> <th>Name</th> <th>Phone</th><th>Email-ID</th> </tr>";
while($row = mysql_fetch_array( $result ))
 {
 // Print out the contents of each row into a table
 echo "<tr><td>";
 echo $row['id'];
 echo "</td><td>";
 echo $row['name'];
 echo "</td><td>";
 echo $row['phone'];
 echo "</td><td>";
 echo $row['emailId'];
 echo "</td></tr>";
}
echo "</table>";
?>
```

Output



Ex. 5.5.2 : ***Create a HTML form "result.html" with a text box and a submit button to accept registration number of the student. Write a "result.php" code to check the status of the result from the table to display whether the student has "PASS" or "FAIL" status. Assume that the MYSQL database "my_db" has the table "result_table" with two columns REG_NO and STATUS. Also write a PHP program to delete a record from result_table.***

Sol. :

Step 1 : Create a database named my_db. Create a table result_table for this database and insert the values in this table. The table is created as follows –

REG_NO	STATUS
101	PASS
102	FAIL
103	PASS
104	FAIL
105	PASS

Step 2 : Create an HTML form to accept the registration number, the HTML document is as follows –

result.html

```
<!DOCTYPE html>
<html>
 <head>
    <title> STUDENT RESULT </title>
 </head>
 <body>
    <form name="myform" method="post" action="http://localhost/php-examples/result.php">
       <input type="text" name="reg_no"/>
       <input type="submit" value="Submit"/>
    </form>
```

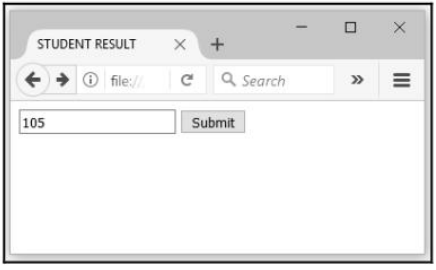
```
</body>
</html>
```

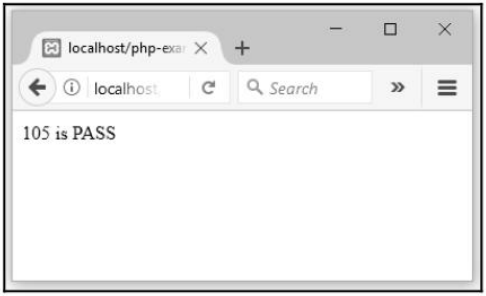
Step 3 : Create a PHP script to accept the registration number. This php script will connect to MYSQL database and the status(PASS or FAIL) of the corresponding registration number will be displayed.

result.php

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost","root","");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("my_db",$conn); //select the database
//Execution of Query for displaying the data
$reg_no = intval($_POST['reg_no']);

$result=mysql_query("SELECT REG_NO,STATUS FROM result_table where REG_NO=$reg_no ");
 while($row = mysql_fetch_array($result))
  {
   echo $row['REG_NO'] . " is " . $row['STATUS'];
   echo "<br />";
  }
mysql_close($conn); //closing the database
?>
```

Step 4 : Load the HTML form created in Step 2 and click the submit button by entering some registration number.





Step 5 : Deletion of record when user submits the registration number.

```
<?php
// Make a MySQL Connection
$conn=mysql_connect("localhost","root","");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("my_db",$conn); //select the database
//Execution of Query for displaying the data
$reg_no = intval($_POST['reg_no']);

$result=mysql_query("DELETE FROM result_table where REG_NO=$reg_no");
 while($row = mysql_fetch_array($result))
  {
   echo $row['REG_NO'] . " is " . $row['STATUS'];          //Each record will be displayed
                                                           //line by line
   echo "<br />";
  }

mysql_close($conn); //closing the database
?>
```

Ex. 5.5.3 : ***Write a user defined function 'CalculateInterest' using PHP to find the simple interest to be paid for a loan amount. Read the loan amount, the number of years and rate of interest from a database table called LOANDETAILS having three fields AMT, YEARS, and RATE, and Calculate the interest using the user defined function.***

Sol. :

Step 1 : Create a database table named LOANDETAILS having three fields AMT, YEARS and RATE. Insert the values in it. The sample table will be as follows –

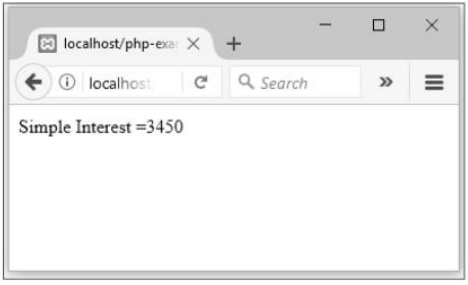
AMT	YEARS	RATE
10000	5	6.9

Step 2 : The PHP code for calculating the simple interest the above values in a function will be as follows –

Interest.php

```
<?php
function CalculateInterest()
{
 // Make a MySQL Connection
 $conn=mysql_connect("localhost","root","");
 if(!$conn)
 {
      die('error in connection'.mysql_error());
 }
 mysql_select_db("my_db",$conn); //select the database
 //Execution of Query for displaying the data
 $result=mysql_query("SELECT * FROM LOANDETAILS");
 $row = mysql_fetch_array($result);
      $amount= $row['AMT'];
      $rate= $row['RATE'];
      $years= $row['YEARS'];
      $interest=($amount * $rate *$years)/100;
 mysql_close($conn); //closing the database
 return $interest;
}
print "Simple Interest =".CalculateInterest();
?>
```

Output



Notes

Laboratory Work

Contents

Laboratory Work

Experiment 1 : a) Install and configure PHP, web server, MYSQL

b) Write a program to print "Welcome to PHP".

c) Write a simple PHP program using expressions and operators.

Sol. : a) Refer Section 1.1.2

b) Refer Section 1.1.1

c) Refer Section 1.2.4

Experiment 2 : Write a PHP program to demonstrate the use of decision making control structures using -

a) If statement

b) If-else statement

c) Switch statement

Sol. : Refer Section 1.3

Experiment 3 : 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

Sol. : Refer Section 1.4

Experiment 4 : Write a PHP program for creating and manipulating -

a) Indexed array

b) Associative array

c) Multidimensional array

Sol. : Refer Section 2.1.2

Experiment 5 : a) Write a PHP program to -

i) Calculate length of string.

ii) 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.

Sol. : Refer Section 2.5

Experiment 6 : Write a simple PHP program to demonstrate use of simple function and parameterized function.

Sol. : Refer Section 2.4

Experiment 7 : Write a simple PHP program to create PDF document by using graphics concepts.

Sol. : Refer Section 2.6.4

Experiment 8 : 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.

Sol. : a) Refer Section 3.3.2

b) Refer Section 3.2

Experiment 9 : Write a simple PHP program on Introspection and Serialization.

Sol. : Refer Section 3.4

Experiment 10 : Design a web page using following form controls :

a) Text box, b) Radio button, c) Check box, d) Buttons

Sol. : Refer Section 4.2

Experiment 11 : Design a web page using following form controls :

a) List box, b) Combo box, c) Hidden field box

Sol. : Refer Section 4.2

Experiment 12 : Develop web page with data validation.

Sol. : Refer Section 4.4

Experiment 13 : Write simple PHP program to -

a) Set cookies and read it.

b) Demonstrate session management.

Sol. : a) Refer Section 4.5

b) Refer Section 4.6

Experiment 14 : Write a simple PHP program for sending and receiving plain text message (e-mail).

Sol. : Refer Section 4.7

Experiment 15 : Develop a simple application to -

a) Enter data into database.

b) Retrieve and present data from database.

Sol. : Refer Section 5.1.1

Experiment 16 : Develop a simple application to Update, Delete table data from database.

Sol. : Refer example 5.1.1

❑❑❑

Solved Sample Test Paper - I

Web Application Development Using PHP

S.Y. Diploma, Semester - VI

(Computer Engineering Group) (CO/CM/IF/CW)

Time : 1 Hour] **[Total Marks : 20**

(1) All questions are compulsory.

(2) Illustrate your answers with neat sketches wherever necessary.

(3) Figures to the right indicate full marks.

(4) Assume suitable data, if necessary.

(5) Preferably, write the answers.

Q.1 ***Attempt any FOUR*** **[8]**

a) *Enlist any four advantages of PHP ?* **(Refer section 1.1)**

b) *Explain how to write a PHP document ?* **(Refer section 1.1.1)**

c) *What are bitwise operators in PHP ?* ***(Refer section 1.2.4)***

d) *Explain the term - Arrays.* **(Refer section 2.1)**

e) *What are actual and formal parameters ?* **(Refer section 2.4.2)**

f) *How to define class in PHP ?* **(Refer section 3.1.1)**

Q.2 ***Attempt any THREE*** **[12]**

a) *Explain data types used in PHP.* **(Refer section 1.2.2)**

b) *Write a PHP program to find the largest number among three numbers.* **(Refer example 1.3.1)**

c) *Explain – if and switch statement.* **(Refer section 1.3)**

d) *How will you create and manipulate arrays ?* **(Refer example 2.1.1)**

e) *Write a short note on – visibility in PHP.* **(Refer section 3.1.5)**

f) *Write a PHP code for creating basic image.* **(Refer section 2.6.1)**

❑❑❑

Solved Sample Test Paper - II

Web Based Application Development Using PHP

S.Y. Diploma, Semester - VI

(Computer Engineering Group) (CO/CM/ IF/CW)

Time : 1 Hour] **[Total Marks : 20**

(1) All questions are compulsory.

(2) Illustrate your answers with neat sketches wherever necessary.

(3) Figures to the right indicate full marks.

(4) Assume suitable data, if necessary.

(5) Preferably, write the answers.

Q.1 ***Attempt any FOUR*** **[8]**

a) *What is meant by inheritance ?* **(Refer section 3.3.2)**

b) *What is data encapsulation ?* **(Refer section 3.3.1)**

c) *What is the difference between GET and POST methods ?* **(Refer section 4.1.2)**

d) *Enlist various form of controls in PHP ?* **(Refer section 4.2)**

e) *Enlist the advantages of MySQL.* **(Refer section 5.1)**

f) *Give MySQL query for creation of student table.* **(Refer section 5.1.1)**

Q.2 ***Attempt any THREE*** **[12]**

a) *Explain the concept of introspection in PHP.* **(Refer section 3.4)**

b) *Write a PHP program to demonstrate function overloading.* **(Refer example 3.3.3)**

c) *Explain the concept of Subglobal Array* **(Refer section 4.1.3)**

d) *How to use radio button in form ? Give suitable example.* **(Refer section 4.2.3)**

e) *Write a program in PHP to insert data in database table.* **(Refer section 5.3)**

f) *Explain – deletion of record from database.* **(Refer section 5.4)**

❑❑❑

Solved Sample Question Paper

Web Based Application Development Using PHP

S.Y. Diploma, Semester - VI

(Computer Engineering Group) (CO/CM/IF/CW)

Time : 3 Hours] **[Total Marks : 70**

INSTRUCTIONS

1. **All questions are compulsory.**
2. **Illustrate your answers with neat sketches wherever necessary.**
3. **Figures to the right indicates full marks.**
4. **Assume suitable data if necessary.**
5. **Preferably write the answers in sequential order.**

Q.1 *Attempt any **FIVE** of the following* **[Marks 10]**

a) *Explain how to use variables in PHP ?* **(Refer section 1.2.1)**

b) *What is constant in PHP ?* **(Refer section 1.2.3)**

c) *How to delete element from an array ?* **(Refer section 2.1.1)**

d) *How to use flip function in PHP ?* **(Refer section 2.2)**

e) *How to define properties and method of a class ? Explain with suitable example.* **(Refer section 3.1)**

f) *Explain the use of textfield in PHP.* **(Refer section 4.2.1)**

g) *Name the function of MySQL which is used to count the number of fields of a table.* **(Refer section 5.3)**

Q.2 *Attempt any **THREE** of the following* **[Marks 12]**

a) *Explain the arithmetic operators that can be used in PHP.* **(Refer section 1.2.4)**

b) *Explain print and echo functions with PHP programs.* **(Refer section 1.2.6)**

c) *Give an example program for traversing an array using foreach programming construct.* **(Refer section 2.3)**

d) *What is the use of anonymous function in PHP? Explain with example.* **(Refer section 2.4.3)**

Q.3 *Attempt any **THREE** of the following* **[Marks 12]**

a) *How to use overriding function in PHP ?* **(Refer section 3.3.4)**

b) *Write a short note on – Cloning object.* **(Refer section 3.3.5)**

c) *Write a PHP program to select your favorite programming language using Checkbox.* **(Refer example 4.2.4)**

d) *Write a PHP program to update data present in the database table.* **(Refer example 5.4)**

Q.4 *Attempt any **THREE** of the following :* **[Marks 12]**

a) *Write a PHP program to display Welcome message.* **(Refer section 1.1.1)**

b) *Explain the associated arrays with example.* **(Refer section 2.1.2)**

c) *How to define and use destructor in PHP ?* **(Refer example 3.2)**

d) *Explain the List control with suitable example.* **(Refer section 4.2.5)**

e) *Write a PHP program to count total number of rows in the database table.* **(Refer section 5.3)**

Q.5 *Attempt any **TWO** of the following :* **[Marks 12]**

a) *Write a PHP program to scale any given image.* **(Refer example 2.6.3)**

b) *Explain bitwise and relational operators in PHP.* **(Refer section 1.2.4)**

c) *Write a PHP program, to demonstrate various string manipulation operations.* **(Refer example 2.5)**

Q.6 *Attempt any **TWO** of the following :* **[Marks 12]**

a) *Write a PHP program to accept a positive integer 'N' through a HTML form and to display the sum of all the numbers from 1 to N.* **(Refer example 4.2.4)**

b) *Create HTML form with one textbox to get user's name. Also write PHP code to show length of entered name when, the HTML form is submitted.* **(Refer example 4.2.1)**

c) *Create a HTML form "result.html" with a text box and a submit button to accept registration number of the student. Write a "result.php" code to check the status of the result from the table to display whether the student has "PASS" or "FAIL" status. Assume that the MYSQL database "my_db" has the table "result_table" with two columns REG_NO and STATUS. Also write a PHP program to delete a record from result_table.* **(Refer example 5.5.2)**

❑❑❑