Unit 5

PHP

Topic

Introduction to PHP, syntax, variables, constants

Topic objective

To discuss about basics of PHP with its syntax, variables and constants.

Introduction to PHP (CO5)

- The term PHP is an acronym for *PHP: Hypertext Preprocessor*. PHP is a server-side scripting language designed specifically for web development.
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP (Lightweight Directory Access Protocol). PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Introduction to PHP (CO5)

- It supports main protocols like HTTP Basic, HTTP Digest, IMAP, FTP, and others.

Common Uses of PHP:

PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them. The other uses of PHP are:

- PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user.
- You add, delete, modify elements within your database thru PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Basics syntax (CO5)

- A PHP script can be placed anywhere in the document.
- A PHP script starts with <?php and ends with ?>:

```
<?php
// PHP code goes here
?>
```

- The default file extension for PHP files is ".php". A PHP file normally contains HTML tags, and some PHP scripting code.
- Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:
- Example-**<html>**

```
<head><title>Hello World</title>
<body><?php echo "Hello, World!";?></body></html
```

Basics syntax (CO5)

Web Server - PHP will work with virtually all Web Server software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server.

Download Apache for free here: http://httpd.apache.org/download.cgi

Database - PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database. Download MySQL for free here:

http://www.mysql.com/downloads/index.html

PHP Parser - In order to process PHP script instructions, a parser must be installed to generate HTML output that can be sent to the Web Browser. This tutorial will guide you how to install PHP parser on your computer.

Variables & Constants (CO5)

- In PHP, variables and constants are used to store and represent data in a program. However, they have some key differences:

Variables:

- Variables are used to store data that may change during the execution of a script.
- The value of a variable can be reassigned multiple times during the program's execution.
- PHP variables start with a dollar sign ($) followed by the variable name (e.g., $variableName).
- Variables are case-sensitive, so $variable and $Variable are considered different variables.
- You don't need to declare variables explicitly before using them; they are created dynamically when assigned a value.

Variables & Constants (CO5)

- $name = “Rakesh"; // String variable
- $age = 30;   // Integer variable
- $height = 1.75; // Float variable
- $isStudent = true; // Boolean variable

<u>Rules for declaring PHP variable:</u>

- A variable must start with a dollar ($) sign, followed by the variable name.
- It can only contain alpha-numeric character and underscore (A-z, 0-9, _).
- A variable name must start with a letter or underscore (_) character.
- A PHP variable name cannot contain spaces.
- One thing to be kept in mind that the variable name cannot start with a number or special symbols.
- PHP variables are case-sensitive, so $name and $NAME both are treated as different variable.

Variables & Constants (CO5)

PHP Variable: Declaring string, integer, and float

Let's see the example to store string, integer, and float values in PHP variables.

```
<?php
$str="hello string";
$x=200;
$y=44.6;
echo "string is: $str <br/>";
echo "integer is: $x <br/>";
echo "float is: $y <br/>";
?>
```

Variables & Constants (CO5)

Output:

```
string is: hello string
integer is: 200
float is: 44.6
```

Variables & Constants (CO5)

PHP constants are name or identifier that can't be changed during the execution of the script except for magic constants, which are not really constants.

PHP constants can be defined by 2 ways:

- Using define() function
- Using const keyword

Constants are similar to the variable except once they defined, they can never be undefined or changed.

Variables & Constants (CO5)

They remain constant across the entire program. PHP constants follow the same PHP variable rules.

For example, it can be started with a letter or underscore only. Conventionally, PHP constants should be defined in uppercase letters.

1. PHP constant: define()

Use the define() function to create a constant.

It defines constant at run time. Let's see the syntax of define() function in PHP.

define(name, value, **case**-insensitive)

```
<?php
define("MESSAGE","Hello PHP");
echo MESSAGE;
?>
```

Output:

```
Hello PHP
```

Variables & Constants (CO5)

2. PHP constant: const keyword

PHP introduced a keyword const to create a constant. The const keyword defines constants at compile time. It is a language construct, not a function. The constant defined using const keyword are case-sensitive.

```
<?php
const MESSAGE="Hello const by PHP";
echo MESSAGE;
?>
```

Output:

```
Hello const by PHP
```

Daily Quiz

1) PHP stands for -

A)Hypertext Preprocessor

B)Pretext Hypertext Preprocessor

C)Personal Home Processor

D)None of the above

2) Who is known as the father of PHP?

A)Drek Kolkevi

B)List Barely

C)Rasmus Lerdrof

D)None of the above

Daily Quiz

3) Which of the following statements is true about variables and constants in PHP?

A) Variables are used for storing data that remains the same throughout the script execution.

B) Constants can be reassigned multiple times during the execution of a script.

C) Variables are case-insensitive, while constants are case-sensitive.

D) Constants are created dynamically when assigned a value, while variables need to be declared before use.

4)Variable name in PHP starts with -

A)! (Exclamation)

B)$ (Dollar)

C)& (Ampersand)

D)# (Hash)

Daily Quiz

5) Which of the following is the default file extension of PHP?

A).php

B).hphp

C).xml

D).html

6) Which of the following is not a variable scope in PHP?

A)Extern

B)Local

C)Static

D)Global

Topic

Overview of Data Type, Operator & Expressions in PHP

Topic objective

To discuss about data type, operator and expression in PHP

Recap

In the last lecture we have discussed about

- Introduction of basic of PHP with its syntax variables and constant.

Data type (CO5)

PHP data types are used to hold different types of data or values. PHP supports 8 primitive data types that can be categorized further in 3 types:

- Scalar Types (predefined)
- Compound Types (user-defined)
- Special Types

1 Scalar Types- It holds only single value. There are 4 scalar data types in PHP.

- boolean
- integer
- float
- string

Data type (CO5)

2 Compound Types- It can hold multiple values. There are 2 compound data types in PHP.

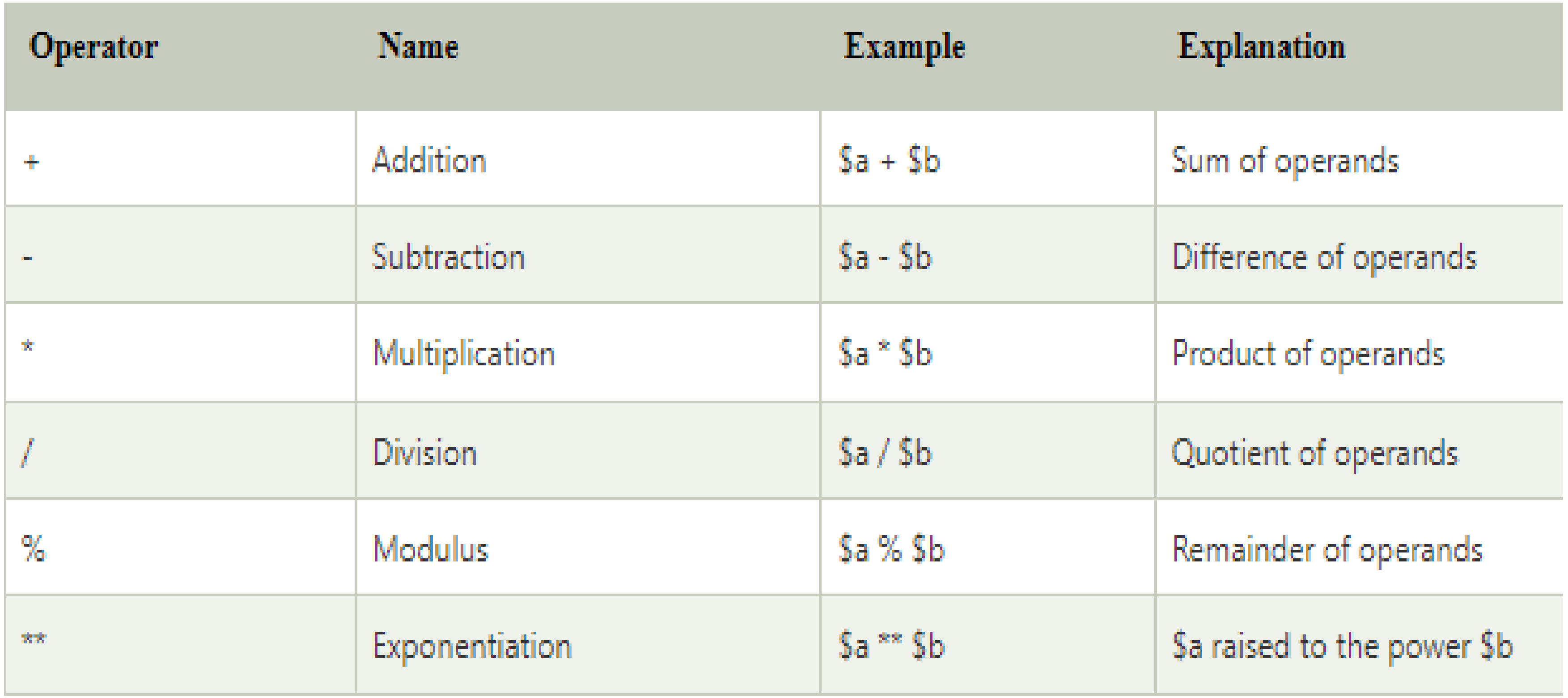
- array
- object

3 Special Types- There are 2 special data types in PHP.

- resource
- NULL

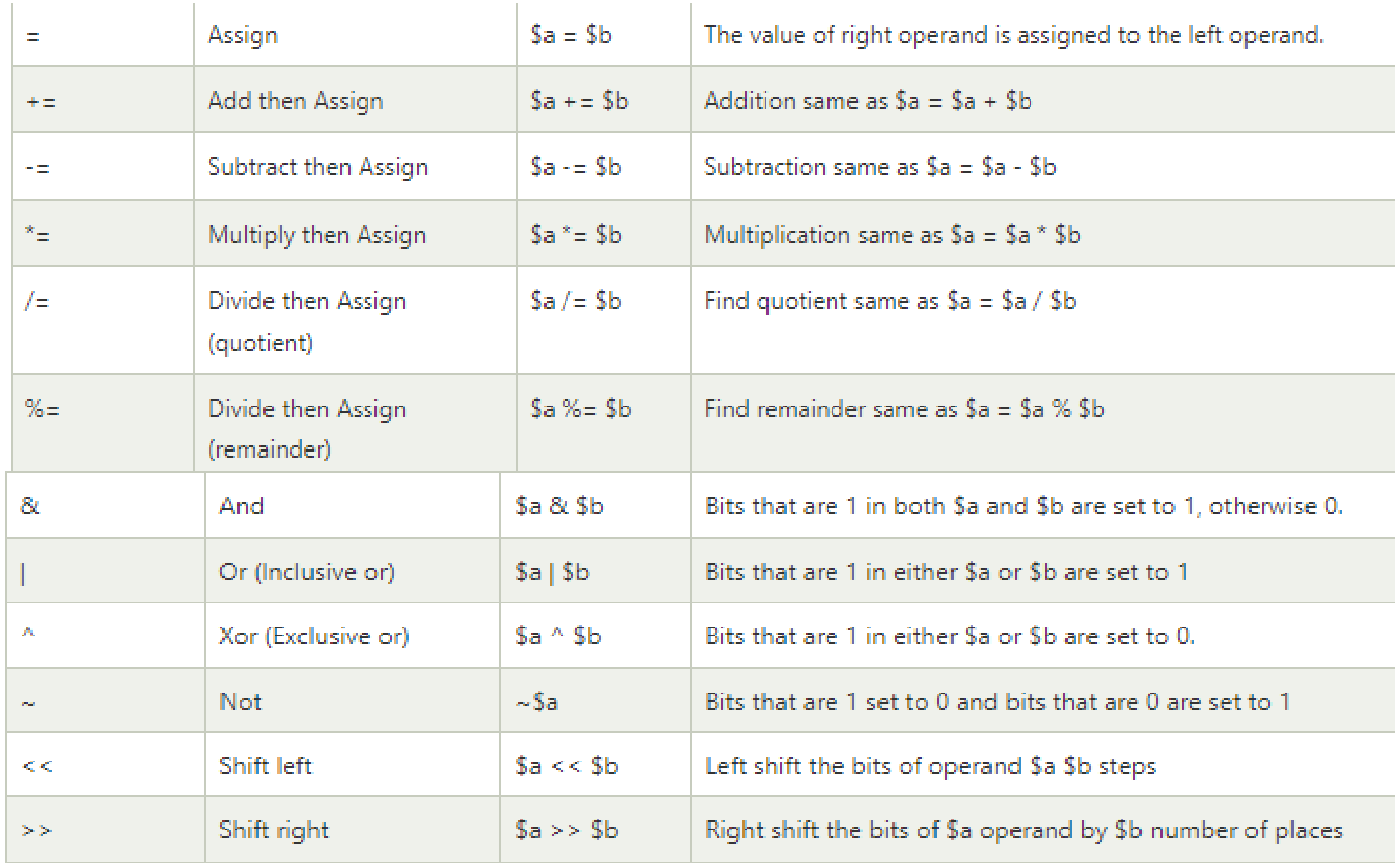
Operator (CO5)

PHP Operator is a symbol i.e used to perform operations on operands. In simple words, operators are used to perform operations on variables or values.



Operator	Name	Example	Explanation
+	Addition	$a + $b	Sum of operands
-	Subtraction	$a - $b	Difference of operands
*	Multiplication	$a * $b	Product of operands
/	Division	$a / $b	Quotient of operands
%	Modulus	$a % $b	Remainder of operands
**	Exponentiation	$a ** $b	$a raised to the power $b

Operator (CO5)



=	Assign	$a = $b	The value of right operand is assigned to the left operand.
+=	Add then Assign	$a += $b	Addition same as $a = $a + $b
-=	Subtract then Assign	$a -= $b	Subtraction same as $a = $a - $b
*=	Multiply then Assign	$a *= $b	Multiplication same as $a = $a * $b
/=	Divide then Assign (quotient)	$a /= $b	Find quotient same as $a = $a / $b
%=	Divide then Assign (remainder)	$a %= $b	Find remainder same as $a = $a % $b
&	And	$a & $b	Bits that are 1 in both $a and $b are set to 1, otherwise 0.
|	Or (Inclusive or)	$a | $b	Bits that are 1 in either $a or $b are set to 1
^	Xor (Exclusive or)	$a ^ $b	Bits that are 1 in either $a or $b are set to 0.
~	Not	~$a	Bits that are 1 set to 0 and bits that are 0 are set to 1
<<	Shift left	$a << $b	Left shift the bits of operand $a $b steps
>>	Shift right	$a >> $b	Right shift the bits of $a operand by $b number of places

Expression (CO5)

- An *expression* is a bit of PHP that can be evaluated to produce a value. The simplest expressions are literal values and variables.
- A literal value evaluates to itself, while a variable evaluates to the value stored in the variable.
- More complex expressions can be formed using simple expressions and operators.

Control flow and Decision making statements (CO5)

1 PHP if- statement allows conditional execution of code. It is executed if condition is true. If statement is used to executes the block of code exist inside the if statement only if the specified condition is true.

- Syntax
  - **if**(condition){
  - //code to be executed
  - }

Control flow and Decision making statements (CO5)

2 PHP if-else statement is executed whether condition is true or false. If-else statement is slightly different from if statement. It executes one block of code if the specified condition is true and another block of code if the condition is false.

- Syntax
  - **if**(condition){
  - //code to be executed if true
  - }**else**{
  - //code to be executed if false
  - }

Control flow and Decision making statements (CO5)

3 The PHP if-else-if is a special statement used to combine multiple if-else statements. So, we can check multiple conditions using this statement.

- Syntax
  - **if** (condition1){
  - //code to be executed if condition1 is true
  - } **elseif** (condition2){
  - //code to be executed if condition2 is true
  - } **elseif** (condition3){
  - //code to be executed if condition3 is true
  - ....
  - } **else**{
  - //code to be executed if all given conditions are false
  - }

Control flow and Decision making statements (CO5)

4 PHP switch statement is used to execute one statement from multiple conditions. It works like PHP if-else-if statement.

- Syntax
  - **switch**(expression){
  - **case** value1:
  - //code to be executed
  - **break**;
  - **case** value2:
  - //code to be executed
  - **break**;
  - ......
  - **default**:
  - code to be executed **if** all cases are not matched;
  - }

Control flow and Decision making statements (CO5)

5 PHP for loop can be used to traverse set of code for the specified number of times. It should be used if the number of iterations is known otherwise use while loop. This means for loop is used when you already know how many times you want to execute a block of code. It allows users to put all the loop related statements in one place.

- Syntax
  - **for**(initialization; condition; increment/decrement){
  - //code to be executed
  - }

Control flow and Decision making statements (CO5)

6 PHP while loop can be used to traverse set of code like for loop. The while loop executes a block of code repeatedly until the condition is FALSE. Once the condition gets FALSE, it exits from the body of loop. It should be used if the number of iterations is not known.

- Syntax
  - **while**(condition){
  - //code to be executed
  - }

Control flow and Decision making statements (CO5)

7 PHP do-while loop can be used to traverse set of code like php while loop. The PHP do-while loop is guaranteed to run at least once. The PHP do-while loop is used to execute a set of code of the program several times. If you have to execute the loop at least once and the number of iterations is not even fixed, it is recommended to use the **do-while** loop.

- Syntax
  - **do**{
  - //code to be executed
  - }**while**(condition);

Control flow and Decision making statements (CO5)

8 PHP break statement breaks the execution of the current for, while, do-while, switch, and for-each loop. If you use break inside inner loop, it breaks the execution of inner loop only. The break statement can be used in all types of loops such as while, do-while, for, foreach loop, and also with switch case.

- Syntax
  - jump statement;
  - **break**;

Daily Quiz

1) Variable name in PHP starts with -

! (Exclamation)

$ (Dollar)

& (Ampersand)

# (Hash)

2) Default file extension of PHP?

A).php

B).hphp

C).xml

D).html

Daily Quiz

3) Which of the following is used to display the output in PHP?

A)echo

B)write

C)print

D)Both (a) and (c)

4) Which of the following is correct to add a comment in php?

A)& …… &

B)// ……

C)/* …… */

D)Both (b) and (c)

Topic

Overview of function, string & arrays in PHP

Topic objective

To discuss about function, string and arrays in PHP

Recap

In the last lecture we have discussed about

- Introduction of basic of PHP with Data Type, Operator & Expressions, Control flow and Decision making statements.

Functions (CO5)

- PHP function is a piece of code that can be reused many times. It can take input as argument list and return value. There are thousands of built-in functions in PHP.
- In PHP, we can define **Conditional function**, **Function within Function** and **Recursive function** also.
- PHP User-defined Functions
- We can declare and call user-defined functions easily. Let's see the syntax to declare user-defined functions.
- Syntax
    - **function** functionname(){
    - //code to be executed
    - }

Functions (CO5)

1 PHP Parameterized functions are the functions with parameters. You can pass any number of parameters inside a function. These passed parameters act as variables inside your function. They are specified inside the parentheses, after the function name. The output depends upon the dynamic values passed as the parameters into the function.

Functions (CO5)

2 PHP call by value function allows you to call function by value and reference both. In case of PHP call by value, actual value is not modified if it is modified inside the function.

- <?php
- **function** adder($str2)
- {
-     $str2 .= 'Call By Value';
- }
- $str = 'Hello ';
- adder($str);
- echo $str;
- ?>

Functions (CO5)

3 In case of PHP call by reference, actual value is modified if it is modified inside the function. In such case, you need to use & (ampersand) symbol with formal arguments. The & represents reference of the variable.

- <?php
- **function** adder(&$str2)
- {
-     $str2 .= 'Call By Reference';
- }
- $str = 'This is ';
- adder($str);
- echo $str;
- ?>

Functions (CO5)

4 PHP Default Argument Values Function allows you to define C++ style default argument values. In such case, if you don't pass any value to the function, it will use default argument value.

- <?php
- **function** sayHello($name="Ram"){
- echo "Hello $name<br/>";
- }
- sayHello("Sonoo");
- sayHello();//passing no value
- sayHello("Vimal");
- ?>

String (CO5)

PHP string is a sequence of characters i.e., used to store and manipulate text. PHP supports only 256-character set and so that it does not offer native Unicode support. There are 4 ways to specify a string literal in PHP.

1 Single Quoted- We can create a string in PHP by enclosing the text in a single-quote. It is the easiest way to specify string in PHP.

For specifying a literal single quote, escape it with a backslash (\) and to specify a literal backslash (\) use double backslash (\\).

All the other instances with backslash such as \r or \n, will be output same as they specified instead of having any special meaning.

String (CO5)

2 Doubled Quoted- In PHP, we can specify string through enclosing text within double quote also.

But escape sequences and variables will be interpreted using double quote PHP strings.

```
<?php
$str="Hello text within double quote";
echo $str;
?>
```

Output:

```
Hello text within double quote
```

NIET Greater Noida GET FUTURE READY Estd 2001 AN AUTONOMOUS INSTITUTE

String (CO5)

3 Heredoc syntax (<<<) is the third way to delimit strings. In Heredoc syntax, an identifier is provided after this heredoc <<< operator, and immediately a new line is started to write any text. To close the quotation, the string follows itself and then again that same identifier is provided. That closing identifier must begin from the new line without any whitespace or tab.

```
<?php
   $str = <<<Demo
It is a valid example
Demo;    //Valid code as whitespace or tab is not valid before closing identifier
echo $str;
?>
Output:
It is a valid example
```

String (CO5)

4 Newdoc is similar to the heredoc, but in newdoc parsing is not done.

- It is also identified with three less than symbols <<< followed by an identifier.
- But here identifier is enclosed in single-quote, e.g. <<<'EXP'. Newdoc follows the same rule as heredocs.
- The difference between newdoc and heredoc is that - Newdoc is a single-quoted string whereas heredoc is a double-quoted string.

Array (CO5)

PHP array is an ordered map (contains value on the basis of key). It is used to hold multiple values of similar type in a single variable. We don't need to define multiple variables. By the help of single loop, we can traverse all the elements of an array. We can sort the elements of array. There are 3 types of array in PHP.

1 PHP indexed array is represented by number which starts from 0. We can store number, string and object in the PHP array.

All PHP array elements are assigned to an index number by default. There are two ways to define indexed array:

Array (CO5)

- 1st way:
- $season=array("summer","winter","spring","autumn");
- 2nd way:
- $season[0]="summer";
- $season[1]="winter";
- $season[2]="spring";
- $season[3]="autumn";

Array (CO5)

2 PHP associative array allows you to associate name/label with each array elements in PHP using => symbol. Such way, you can easily remember the element because each element is represented by label than an incremented number. There are two ways to define associative array:

- 1st way:
- $salary=**array**("Sonoo"=>"550000","Vimal"=>"250000","Ratan"=>"200000");
- 2nd way:
- $salary["Sonoo"]="550000";
- $salary["Vimal"]="250000";
- $salary["Ratan"]="200000";

Array (CO5)

3 PHP multidimensional array is also known as array of arrays. It allows you to store tabular data in an array. PHP multidimensional array can be represented in the form of matrix which is represented by row * column.

- $emp = **array**
- (
- **array**(1,"sonoo",400000),
- **array**(2,"john",500000),
- **array**(3,"rahul",300000)
-);

Daily Quiz

1) How to define a function in PHP?

a) functionName(parameters) {function body}

b) function {function body}

c) function functionName(parameters) {function body}

d) data type functionName(parameters) {function body}

2) Which is the right way of declaring a variable in PHP?

a) $3hello

b) $_hello

c) $this

d) $5_Hello

Topic

Overview of files and directories

Topic objective

To discuss about Understanding file& directory, Opening and closing, a file, Coping, renaming and deleting a file, working with directories, Creating and deleting folder, File Uploading & Downloading in PHP

Understanding file & directory, working with directories (CO5)

- Now that we have spent some time looking at how to work with files in PHP it is now time to look at how to work with file system directories.
- PHP provides a number of functions that can be used to perform tasks such as identifying and changing the current directory, creating new directories, deleting existing directories and listing the contents of a directory.

1 Creating Directories in PHP- A new directory can be created in PHP using the mkdir() function. This function takes a path to the directory to be created.

- To create a directory in the same directory as your PHP script simply provide the directory name. To create a new directory in a different directory specify the full path when calling mkdir().
- A second, optional argument allows the specification of permissions on the directory (controlling such issues as whether the directory is writable):

```
<?php
$result = mkdir ("/path/to/directory", "0777");
?>
```

Understanding file & directory, working with directories (CO5)

2 Deleting a directory- Directories are deleted in PHP using the *rmdir()* function. *rmdir()* takes a single argument, the name of the directory to be deleted.

- The deletion will only be successful if the directory is empty. If the directory contains files or other sub-directories the deletion cannot be performed until those files and sub-directories are also deleted.

3 To Close a directory- We use closedir() function in order to close a directory after reading its contents.

Syntax:

```
$dir_handle = opendir($dir_path);
...
...
closedir($dir_handle);
```

Understanding file & directory, working with directories (CO5)

4 To open a directory- The opendir() function in PHP is an inbuilt function which is used to open a directory handle. The path of the directory to be opened is sent as a parameter to the opendir() function and it returns a directory handle resource on success, or FALSE on failure. The opendir() function is used to open up a directory handle to be used in subsequent with other directory functions such as closedir(), readdir(), and rewinddir().

Syntax:

```
opendir($path, $context)
```

Understanding file & directory, working with directories (CO5)

5 To copy a directory- The copy() function is used to make a copy of a specified file. It makes a copy of the source file to the destination file and if the destination file already exists, it gets overwritten. The copy() function returns true on success and false on failure.

Syntax:

Bool copy($source, $dest)

6 To rename a directory- The rename() function in PHP is an inbuilt function which is used to rename a file or directory. It makes an attempt to change an old name of a file or directory with a new name specified by the user and it may move between directories if necessary.

Understanding file & directory, working with directories (CO5)

If the new name specified by the user already exists, the rename() function overwrites it. The old name of the file and the new name specified by the user are sent as parameters to the rename() function and it returns True on success and a False on failure.

Syntax:

rename(oldname, newname, context)

Creating and deleting folder (CO5)

- Earlier we have seen about deleting just files which are present in a directory. Now, What if you have sub directories with files inside the folder.
- In that case the above script won't able to work. You'll need to create recursive function to delete all files, sub-directories and parent directory altogether.
- Simply deleting a folder with using php's function rmdir() won't work. It will throw some exceptions if you attempt to remove folder directly with files in it.
- So first of all you have to delete files one by one from each sub folder and then delete folders by removing parent folder.

Creating and deleting folder (CO5)

```
<?php
// delete all files and sub-folders from a folder
function deleteAll($dir) {
foreach(glob($dir . '/*') as $file) {
if(is_dir($file))
deleteAll($file);
else
unlink($file);
}
rmdir($dir);
}
?>
```

File Uploading & Downloading (CO5)

Uploading- With PHP, it is easy to upload files to the server. However, with ease comes danger, so always be careful when allowing file uploads and follow these steps.

1. Configure The "php.ini" File
2. Create The HTML Form
3. Create The Upload File PHP Script
4. Check if File Already Exists
5. Limit File Size
6. Limit File Type
7. Complete Upload File PHP Script

File Uploading & Downloading (CO5)

PHP Download File- PHP enables you to download file easily using built-in readfile() function. The readfile() function reads a file and writes it to the output buffer.

Syntax

```
int readfile ( string $filename [, bool $use_include_path = false [, resource $context ]] )
```

$filename: represents the file name

$use_include_path: it is the optional parameter. It is by default false. You can set it to true to the search the file in the included_path.

$context: represents the context stream resource.

int: it returns the number of bytes read from the file.

Introduction to Session Control (CO5)

When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.

So; Session variables hold information about one single user, and are available to all pages in one application.

Introduction to Session Control (CO5)

A session is started with the session_start() function. Session variables are set with the PHP global variable: $_SESSION. Now, let's create a new page called "demo_session1.php". In this page, we start a new PHP session and set some session variables:

Example

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
```

Introduction to Session Control (CO5)

```
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
</body>
</html>
```

Session Functions and Functionality (CO5)

- session_abort — Discard session array changes and finish session
- session_cache_expire — Get and/or set current cache expire
- session_cache_limiter — Get and/or set the current cache limiter
- session_commit — Alias of session_write_close
- session_create_id — Create new session id
- session_decode — Decodes session data from a session encoded string
- session_destroy — Destroys all data registered to a session
- session_encode — Encodes the current session data as a session encoded string
- session_gc — Perform session data garbage collection
- session_get_cookie_params — Get the session cookie parameters
- session_id — Get and/or set the current session id
- session_module_name — Get and/or set the current session module
- session_name — Get and/or set the current session name

Session Functions and Functionality (CO5)

- session_name — Get and/or set the current session name
- session_regenerate_id — Update the current session id with a newly generated one
- session_register_shutdown — Session shutdown function
- session_reset — Re-initialize session array with original values
- session_save_path — Get and/or set the current session save path
- session_set_cookie_params — Set the session cookie parameters
- session_set_save_handler — Sets user-level session storage functions
- session_start — Start new or resume existing session
- session_status — Returns the current session status
- session_unset — Free all session variables
- session_write_close — Write session data and end session

Cookie (CO5)

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

Create Cookies With PHP

A cookie is created with the setcookie() function.

Syntax

```
setcookie(name, value, expire, path, domain, secure, httponly);
```

Setting Cookie with PHP (CO5)

Setting Cookie In PHP: To set a cookie in PHP, the setcookie() function is used. The setcookie() function needs to be called prior to any output generated by the script otherwise the cookie will not be set.

Syntax:

setcookie(name, value, expire, path, domain, security);

Parameters: The setcookie() function requires six arguments in general which are:

Name: It is used to set the name of the cookie.

Value: It is used to set the value of the cookie.

Setting Cookie with PHP (CO5)

Expire: It is used to set the expiry timestamp of the cookie after which the cookie can’t be accessed.

Path: It is used to specify the path on the server for which the cookie will be available.

Domain: It is used to specify the domain for which the cookie is available.

Security: It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists.

Setting Cookie with PHP (CO5)

PHP sessions can use cookies depending on how you configure them. Have a look at these settings:

- session.use_cookies (boolean): specifies whether the module will use cookies to store the session id on the client side. Defaults to 1 (enabled).
- session.use_only_cookies (boolean): specifies whether the module will only use cookies to store the session id on the client side. Enabling this setting prevents attacks involved passing session ids in URLs. This setting was added in PHP 4.3.0. Defaults to 1 (enabled) since PHP 5.3.0.

If you disable session cookies, a GET parameter is used instead.

Deleting Cookies (CO5)

PHP sessions can use cookies depending on how you configure them. Have a look at these settings:

- session.use_cookies (boolean): specifies whether the module will use cookies to store the session id on the client side. Defaults to 1 (enabled).
- session.use_only_cookies (boolean): specifies whether the module will only use cookies to store the session id on the client side. Enabling this setting prevents attacks involved passing session ids in URLs. This setting was added in PHP 4.3.0. Defaults to 1 (enabled) since PHP 5.3.0.

If you disable session cookies, a GET parameter is used instead.

Deleting Cookies (CO5)

Use the setcookie() method to delete the cookies. For that, we need to keep the expiry date of the past. We can use the isset() function to check if the cookie has been set before deleting the cookie.

For example, use the $_COOKI[$cookie_name] variable in the isset() function to check if the above created cookie exist.

In the if block, use the setcookie() function. Inside the function, set $cookie_name as the first parameter.

```
if(isset($_COOKIE[$cookie_name])) {
 setcookie($cookie_name, "", time()-3600);
}
```

Deleting Cookies (CO5)

We wrote an empty string and time()-3600 for the expiry time for the second parameter.

As the time() function returns the current time in seconds since Epoch, 3600 subtracted from it will return the past time. Finally, the cookie with the $cookie_name is deleted. We can also use $cookie_name and null value in the setcookie() function to delete the cookies.

```
if(isset($_COOKIE[$cookie_name])) {
 setcookie($cookie_name, null);
}
```

Thus, we can use the setcookie() function to delete cookies in PHP.

Registering Session variables (CO5)

- The PHP session is required so that you can store the user information and use it on different pages of the browser.
- It creates a session with the name or any other useful information you want to store and access on different pages.
- Even after your page is closed you can access the information until the browser does not close.
- This is an important thing to understand if a browser is closed then the session is automatically destroyed.

Registering Session variables (CO5)

- We can create the session by writing session_start()_and destroy the session by using session_destroy(). You can access the session variable by writing $_session[“name”].

```
<?php
  // Starting session
   session_start();
  // Use of session_register() is deprecated
   $username = "PhpScots";
   session_register("username");
  // Use of $_SESSION is preferred
   $_SESSION["username"] = "PhpScots";
?>
```

Destroying the variables and Session **(CO5)**

Destroying the Variable- The unset() function in PHP resets any variable. If unset() is called inside a user-defined function, it unsets the local variables.

If a user wants to unset the global variable inside the function, then he/she has to use $GLOBALS array to do so. The unset() function has no return value.

Destroying the variables and Session **(CO5)**

Destroying the Session- A PHP session can be destroyed by session_destroy() function. This function does not need any argument and a single call can destroy all the session variables. If you want to destroy a single session variable then you can use unset() function to unset a session variable. Here is the example to unset a single variable –

- <?php
- unset($_SESSION['counter']);
- ?>
- Here is the call which will destroy all the session variables –

- <?php
- session_destroy();
- ?>