Que 1. What is constant variable? Explain with example.

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

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

- Using const 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 Minhaz Lariya";
        echo MESSAGE;
?>
```

Que 2. Differentiate include and require.

include()	require()
The **include()** function does not stop the execution of the script even if any error occurs.	The **require()** function will stop the execution of the script when an error occurs.
The **include()** function does not give a fatal error.	The **require()** function gives a fatal error
The **include()** function is mostly used when the file is not required and the application should continue to execute its process when the file is not found.	The **require()** function is mostly used when the file is mandatory for the application.
The **include()** function will only produce a warning (E_WARNING) and the script will continue to execute.	The **require()** will produce a fatal error (E_COMPILE_ERROR) along with the warning.

Que 3. How to declare global variables.

Global variables refer to any variable that is defined outside of the function. Global variables can be accessed from any part of the script i.e. inside and outside of the function. So, a global variable can be declared just like other variable but it must be declared outside of function definition.

Syntax:
$variable_name = data;

Using global keyword
Using array GLOBALS[var_name]: It stores all global variables in an array called $GLOBALS[var_name]. Var_name is the name of the variable. This array is also accessible from within functions and can be used to perform operations on global variables directly.

```php
<?php
    $x = "Minhaz";
    $y = "Gulambhai";
    $z = "Lariya";
    $a = 5;
    $b = 10;

    function concatenate() {
        // Using global keyword
        global $x, $y, $z;
        return $x.$y.$z;
    }

    function add() {
        // Using GLOBALS['var_name']
        $GLOBALS['b'] = $GLOBALS['a'] + $GLOBALS['b'];
    }

    // Print result
    echo concatenate();
    echo"\n";
    add();
    echo $b;
?>
```

Que 4. What is WSGI?

Flask is a WSGI application. A WSGI server is used to run the application, converting incoming HTTP requests to the standard WSGI environ, and converting outgoing WSGI responses to HTTP responses.

Que 5. Explain redirect () method.

A redirect is used in the Flask class to send the user to a particular URL with the status code.
conversely, this status code additionally identifies the issue.
When we access a website, our browser sends a request to the server, and the server replies with what is known as the HTTP status code, which is a three-digit number, The different reasons for errors are:

- Unauthorized access or poor request.
- Unsupported media file types.
- Overload of the backend server.
- Internal hardware/connection error.

Syntax of Redirect
flask.redirect(location,code=302)

Parameters:
location(str): the location which URL directs to.
code(int): The status code for Redirect.
Code: The default code is 302 which means that the move is only temporary.
Return: The response object and redirects the user to another target location with the specified code.

Que 6. Explain OS module's "isdir ()" method.

os.path.isdir() method in Python is used to check whether the specified path is an existing directory or not.
This method follows symbolic link, that means if the specified path is a symbolic link pointing to a directory then the method will return True.

Syntax:
os.path.isdir(path)

Parameter:
path: A path-like object representing a file system path.
Return Type: This method returns a Boolean value of class bool. This method returns True if specified path is an existing directory, otherwise returns False.

Que 7. Who was the father of PHP? Write some basic history.

- Rasmus Lerdorf is often referred to as the "Father of PHP." He created PHP (Hypertext Preprocessor) in 1994, initially as a set of tools for tracking visits to his online resume.
- Over time, PHP evolved into a widely-used server-side scripting language, integral to web development and dynamic websites.
- The acronym "PHP" originally stood for "Personal Home Page."
- However, as PHP evolved and became more powerful, it was renamed to "PHP: Hypertext Preprocessor," reflecting its expanded capabilities in handling dynamic web content.

Que 8. Differentiate print and echo.

S.No.	Echo Statement	Print Statement
1.	In Echo, we can pass multiple arguments separated by commas.	In Print, we cannot pass multiple arguments.
2.	In Echo, we can exhibit the outputs of one or more strings separated by commas.	Through the Print statement, we can only show the strings.
3	Echo can be used with or without parentheses	Print can also be used with or without parentheses.
4.	It never returns any value.	It always returns the integer value that is 1.
5.	This statement is fast as compared to the print statement.	It is slow as compared to the echo statement.

Que 9. How to use Session in Flask?

- Flask-Session is an extension for Flask that supports Server-side Session to your application.
- The Session is the time between the client logs in to the server and logs out of the server.
- The data that is required to be saved in the Session is stored in a temporary directory on the server.
- The data in the Session is stored on the top of cookies and signed by the server cryptographically.
- Each client will have their own session where their own data will be stored in their session.

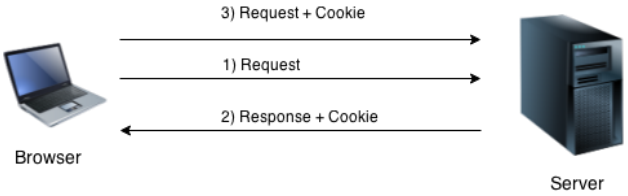
Uses of Session

- Remember each user when they log in
- Store User-specific website settings (theme)
- Store E-Commerce site user items in the cart

Que 1. Explain cookies in detail.

PHP cookie is a small piece of information which is stored at client browser. It is used to recognize the user.

Cookie is created at server side and saved to client browser. Each time when client sends request to the server, cookie is embedded with request. Such way, cookie can be received at the server side.



PHP setcookie() function

PHP setcookie() function is used to set cookie with HTTP response. Once cookie is set, you can access it by $_COOKIE superglobal variable.

Syntax

bool setcookie (string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]])

For E.g

setcookie("CookieName", "CookieValue");/* defining name and value only*/
setcookie("CookieName", "CookieValue", time()+1*60*60);//using expiry in 1 hour(1*60*60 seconds or 3600 seconds)
setcookie("CookieName", "CookieValue", time()+1*60*60, "/mypath/", "mydomain.com", 1);

PHP $_COOKIE

PHP $_COOKIE superglobal variable is used to get cookie.

Example

$value=$_COOKIE["CookieName"];//returns cookie value

$_COOKIE

PHP $_COOKIE superglobal variable is used to get cookie.

Example

$value=$_COOKIE["CookieName"];//returns cookie value

Delete Cookie

If you set the expiration date in past, cookie will be deleted.

File: cookieProgram.php

```
<?php
        setcookie ("CookieName", "", time() - 3600);// set the expiration date to one hour ago
?>
```

Que 2. What is static? Explain with example.

The static keyword is used to declare properties and methods of a class as static. Static properties and methods can be used without creating an instance of the class.

The static keyword is also used to declare variables in a function which keep their value after the function has ended.

- A static class in PHP is a type of class which is instantiated only once in a program.
- It must contain a static member (variable) or a static member function (method) or both.
- The variables and methods are accessed without the creation of an object, using the scope resolution operator(::).
- But here is a catch, that the static method cannot access the non-static variables because that will require the creation of the object first. So, to access variables of a static class we must declare them as static using keyword static.

```php
<?php
    class Class_name {

        // Static variable and static function
        // using static keyword
        public static $var = "text";

        public static function func() {
            echo self::$var;
        }
    }

    Class_name::func();
?>
```

Output: text

Example 2: This example checks if a string created has a length greater than or equal to 7 or not.

```php
<?php

    class GFG
    {
        // Static variable
        public static $num1 = 7;

        // Static function
        public static function check($var)
        {
            // Accessing static variable using
            // self keyword
            if(strlen($var) >= self::$num1)
                return true;
            else
                return false;
        }
    }

    // String is created
    $str = "Lariya Minhaz";

    // Static function is called
    // using scope resolution operator
    if(GFG::check($str))
        echo "String is valid!";
    else
        echo "String is NOT valid!";

?>
```

Output: String is valid!

Que 3. How to send email using PHP?

PHP must be configured correctly in the php.ini file with the details of how your system sends email. Open php.ini file available in /etc/ directory and find the section headed [mail function].

The configuration for Windows should look something like this −

```
[mail function]
; For Win32 only.
SMTP = smtp.secureserver.net

; For win32 only
sendmail_from = webmaster@tutorialspoint.com
```

Sending plain text email

PHP makes use of mail() function to send an email. This function requires three mandatory arguments that specify the recipient's email address, the subject of the the message and the actual message additionally there are other two optional parameters.

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

Sr.No	Parameter & Description
1	**to** Required. Specifies the receiver / receivers of the email
2	**subject** Required. Specifies the subject of the email. This parameter cannot contain any newline characters
3	**message** Required. Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters
4	**headers** Optional. Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\r\n)
5	**parameters** Optional. Specifies an additional parameter to the send mail program

```php
<?php
    $to = "somebody@example.com";
    $subject = "My subject";
    $txt = "Hello world!";
    $headers = "From: webmaster@example.com" . "\r\n" ."CC: somebodyelse@example.com";

    mail($to,$subject,$txt,$headers);
?>
```

Que 1. Explain escapeshellcmd(), shell_exec() method.

escapeshellcmd() escapes any characters in a string that might be used to trick a shell command into executing arbitrary commands.

This function should be used to make sure that any data coming from user input is escaped before this data is passed to the exec() or system() functions, or to the backtick operator.

Following characters are preceded by a backslash: &#;`|*?~<>^()[]{}$\, \x0A and \xFF. ' and " are escaped only if they are not paired. On Windows, all these characters plus % and ! are preceded by a caret (^).

```php
<?php
    $e = escapeshellcmd($userinput);

    // here we don't care if $e has spaces
    system("echo $e");
    $f = escapeshellcmd($filename);

    // and here we do, so we use quotes
    system("touch \"/tmp/$f\"; ls -l \"/tmp/$f\"");
?>
```

The shell_exec() function is an inbuilt function in PHP which is used to execute the commands via shell and return the complete output as a string. The shell_exec is an alias for the backtick operator, for those used to *nix. If the command fails return NULL and the values are not reliable for error checking.

Syntax:
string shell_exec($cmd)
Parameters: This function accepts single parameter $cmd which is used to hold the command that will be executed.
Return Value: This function returns the executed command or NULL if an error occurred.
Note: This function is disabled when PHP is running in safe mode.

```
<?php
$output = shell_exec('ls');
echo "<pre>$output</pre>";
?>
```

Output:
gfg.php
index.html
geeks.php

Que 2. Explain PHP Array and its type in detail.

Arrays in PHP is a type of data structure that allows us to store multiple elements of similar data type under a single variable thereby saving us the effort of creating a different variable for every data. The arrays are helpful to create a list of elements of similar types, which can be accessed using their index or key.

There are basically three types of arrays in PHP:
Indexed or Numeric Arrays: An array with a numeric index where values are stored linearly.
Associative Arrays: An array with a string index where instead of linear storage, each value can be assigned a specific key.
Multidimensional Arrays: An array which contains single or multiple array within it and can be accessed via multiple indices.

Indexed or Numeric Arrays
These type of arrays can be used to store any type of elements, but an index is always a number. By default, the index starts at zero. These arrays can be created in two different ways as shown in the following example:

```php
<?php

    // One way to create an indexed array
    $name_one = array("Zack", "Anthony", "Ram", "Salim", "Raghav");

    // Accessing the elements directly
    echo "Accessing the 1st array elements directly:\n";
    echo $name_one[2], "\n";
    echo $name_one[0], "\n";
    echo $name_one[4], "\n";

    // Second way to create an indexed array
    $name_two[0] = "ZACK";
    $name_two[1] = "ANTHONY";
    $name_two[2] = "RAM";
    $name_two[3] = "SALIM";
    $name_two[4] = "RAGHAV";

    // Accessing the elements directly
    echo "Accessing the 2nd array elements directly:\n";
    echo $name_two[2], "\n";
    echo $name_two[0], "\n";
    echo $name_two[4], "\n";
?>
```

Associative Arrays:
These types of arrays are similar to the indexed arrays but instead of linear storage, every value can be assigned with a user-defined key of string type.

```php
<?php
    // One way to create an associative array
    $name_one = array("Zack"=>"Zara", "Anthony"=>"Any",
                      "Ram"=>"Rani", "Salim"=>"Sara",
                      "Raghav"=>"Ravina");

    // Second way to create an associative array
    $name_two["zack"] = "zara";
    $name_two["anthony"] = "any";
    $name_two["ram"] = "rani";
    $name_two["salim"] = "sara";
    $name_two["raghav"] = "ravina";

    // Accessing the elements directly
    echo "Accessing the elements directly:\n";
    echo $name_two["zack"], "\n";
    echo $name_two["salim"], "\n";
    echo $name_two["anthony"], "\n";
    echo $name_one["Ram"], "\n";
    echo $name_one["Raghav"], "\n";
?>
```

Multidimensional Arrays

Multi-dimensional arrays are such arrays that store another array at each index instead of a single element. In other words, we can define multi-dimensional arrays as an array of arrays. As the name suggests, every element in this array can be an array and they can also hold other sub-arrays within. Arrays or sub-arrays in multidimensional arrays can be accessed using multiple dimensions.

```php
<?php
    // Defining a multidimensional array
    $favorites = array(
        array(
            "name" => "Dave Punk",
            "mob" => "5689741523",
            "email" => "davepunk@gmail.com",
        ),
        array(
            "name" => "Monty Smith",
            "mob" => "2584369721",
            "email" => "montysmith@gmail.com",
        ),
        array(
            "name" => "John Flinch",
            "mob" => "9875147536",
            "email" => "johnflinch@gmail.com",
        )
    );

    // Accessing elements
    echo "Dave Punk email-id is: " . $favorites[0]["email"], "\n";
    echo "John Flinch mobile number is: " . $favorites[2]["mob"];

?>
```

Que 3. Explain exception handling.

An exception is an unexpected program result that can be handled by the program itself. Exception Handling in PHP is almost similar to exception handling in all programming languages.

PHP provides the following specialized keywords for this purpose.
try: It represents a block of code in which exceptions can arise.
catch: It represents a block of code that will be executed when a particular exception has been thrown.
throw: It is used to throw an exception. It is also used to list the exceptions that a function throws, but doesn't handle itself.
finally: It is used in place of a catch block or after a catch block basically it is put for cleanup activity in PHP code.

Why Exception Handling in PHP?

The following are the main advantages of exception handling over error handling

- Separation of error handling code from normal code: In traditional error handling code there is always an if-else block to handle errors. These conditions and code to handle errors got mixed so that became unreadable. With try Catch block code becomes readable.
- Grouping of error types: In PHP both basic types and objects can be thrown as exceptions. It can create a hierarchy of exception objects, group exceptions in namespaces or classes, and categorize them according to types.

```php
<?php
    function demo($var) {
        echo " Before try block";
        try {
            echo "\n Inside try block";
            // If var is zero then only if will be executed
            if($var == 0)
            {
                // If var is zero then only exception is thrown
                throw new Exception('Number is zero.');

                // This line will never be executed
                echo "\n After throw (It will never be executed)";
            }
        }

        // Catch block will be executed only
        // When Exception has been thrown by try block
        catch(Exception $e) {
                echo "\n Exception Caught", $e->getMessage();
            }
            // This line will be executed whether
            // Exception has been thrown or not
            echo "\n After catch (will be always executed)";
    }
    // Exception will not be raised
    demo(5);
    // Exception will be raised here
    demo(0);
?>
```

Que 1. Explain following Functions with example (Any seven)

getdate()

The getdate() function is an inbuilt function in PHP which is used to get date/time information of the current local date/time.

Syntax:

getdate($timestamp)

```
<?php
    print_r(getdate());
?>
```

```
Array
(
   [seconds] => 40
   [minutes] => 25
   [hours] => 6
   [mday] => 3
   [wday] => 2
   [mon] => 7
   [year] => 2018
   [yday] => 183
   [weekday] => Tuesday
   [month] => July
   [0] => 1530599140
)
```

asort()

The asort() function is an inbuilt function in PHP which is used to sort an array according to values. It sorts in a way that relation between indices and values is maintained. By default it sorts in ascending order of values.

Syntax:

bool asort($array, $sorting_type)

```php
<?php
    $arr = array("0" => "Web Technology",
                 "1" => "Machine Learning",
                 "2" => "GeeksforGeeks",
                 "3" => "Computer Graphics",
                 "4" => "Videos",
                 "5" => "Report Bug",
                 "6" => "Article",
                 "7" => "Sudo Placement",
                 "8" => "SContribute",
                 "9" => "Reset",
                 "10" => "Copy",
                 "11" => "IDE",
                 "12" => "Gate Note",
             );

    asort($arr);

    foreach ($arr as $key => $val) {
        echo "[$key] = $val";
        echo"\n";
    }
?>
```

Output

```
[6] = Article
[3] = Computer Graphics
[10] = Copy
[12] = Gate Note
[2] = GeeksforGeeks
[11] = IDE
[1] = Machine Learning
[5] = Report Bug
[9] = Reset
[8] = SContribute
[7] = Sudo Placement
[4] = Videos
[0] = Web Technology
```

define()

The define() function is basically used by programmers to create constant. Constants in PHP are very similar to variables and the only difference between both are the values of constants can not be changed once it is set in a program.

define() returns a Boolean value. It will return TRUE on success and FALSE on failure of the expression.

	Syntax: **define(string $constant, mixed $value, bool $case_insensitive);** <?php define("GREETINGS", "Hello GFG.", true); echo GREETINGS; ?> **Output: Hello GFG.**
floor()	PHP provides a built-in function floor() to get this task done through our PHP script. The floor() function in PHP rounds down the float value to the next lowest integer value. **Syntax:** **float floor(value)** Input : floor(1.9) Output : 1 Input : floor(-1.8) Output : -2 Input : floor(4) Output : 4

fclose()	The fclose() function in PHP is an inbuilt function which is used to close a file which is pointed by an open file pointer. The fclose() function returns true on success and false on failure. It takes the file as an argument which has to be closed and closes that file.
	
	Syntax:
	bool fclose($file)
	
	<?php
	// opening a file using fopen() function
	$check = fopen("gfg.txt", "r");
	
	// closing a file using fclose() function
	fclose($check);
	?>
	Output: true

header()	The header() function is an inbuilt function in PHP which is used to send a raw HTTP header.
	The HTTP functions are those functions which manipulate information sent to the client or browser by the Web server, before any other output has been sent.
	Syntax:
	void header($header, $replace = TRUE, $http_response_code)
	
	<?php
	// PHP program to describes header function
	
	// Redirect the browser
	header("Location: https://www.minhazlariya.org");
	
	// The below code does not get executed
	// while redirecting
	exit;
	?>
	
	Output: This will change location of header, i.e. redirect to the URL

ltrim()	The ltrim() function is a built-in function in PHP which removes whitespaces or other characters (if specified) from the left side of a string. Syntax: ltrim($string, $charlist) Input : $string = " Geeks for Geeks" Output : Geeks for Geeks Input : $string = "!!! ((!!)) Geeks for Geeks", $charlist = "! ()" Output : Geeks for Geeks

substr()

The substr() is a built-in function in PHP that is used to extract a part of string.

Syntax:
substr(string_name, start_position, string_length_to_cut)

```
<?php

// PHP program to illustrate substr()
function Substring($str){
        $len = strlen($str);
        echo substr($str, 8), "\n";
        echo substr($str, 5, $len), "\n";
        echo substr($str, -5, 10), "\n";
        echo substr($str,-8, -5), "\n";
}

// Driver Code
$str="GeeksforGeeks";
Substring($str);
?>
```

Output:
Geeks
forGeeks
Geeks
for

setcookie()

The setcookie() function is used to create a cookie. The setcookie() function defines a cookie to be sent along with other HTTP headers. The setcookie() function should be appeared before the <html> and <head> tag.

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

```
<?php
$value = 'Arecookiesset';

setcookie("TestCookie", $value);
setcookie("check","are cookies set")
?>
<?php
// Print an individual cookie
// echo $_COOKIE["TestCookie"] ."\n";
// echo $_COOKIE["check"] ."\n";

// Another way to debug/test is to view all cookies
print_r($_COOKIE);
?>
```

Output: Array ([TestCookie] => Arecookiesset [check] => are cookies set)

Que 1. Write a program which executes php script into python.

FileName: PythonFile.py

Print("Hello this is Calling from Python File")

```
<?php
  $command_exec = escapeshellcmd(' PythonFile .py');
  $str_output = shell_exec($command_exec);
  echo $str_output;
?>
```

The right privileges need to be given so that the python script is successfully executed.

Note − While working on a Unix type of platform, PHP code is executed as a web user. Hence, the web user should be given the necessary rights to the directories and sub-directories.

Que 2. Explain walk method in detail.

OS.walk() generate the file names in a directory tree by walking the tree either top-down or bottom-up. For each directory in the tree rooted at directory top (including top itself), it yields a 3-tuple (dirpath, dirnames, filenames).

root : Prints out directories only from what you specified.
dirs : Prints out sub-directories from root.
files : Prints out all files from root and directories.

```
# Driver function
import os
if __name__ == "__main__":
for (root,dirs,files) in os.walk('.', topdown=True):
print (root)
print (dirs)
print (files)
print ('--------------------------------')
```

Que 3. Write a PHP script which demonstrate connection and any 2 basic operation functionality (add/update/delete/ search records) with your choice of database.

```
<form method="post">
    <button name="btnInsert" id="btnInsert">Save</button>
    <button name="btnDelete" id="btnDelete">Delete</button>
</form>


<?php
    $Con = mysqli_connect("localhost", "root", "", "MYDB");
    if (isset(_POST["btnInsert"]))
    {
        $Query = "Insert into Student_MST values (1, 'Lariya Minhaz', '9854785478')";
        mysqli_query($Con, $Query);
        echo "Successfully Inserted";
    }

    if (isset(_POST["btnDelete"]))
    {
        Query = "Delete from Student_MST where StudId = 1";
        mysqli_query($Con, $Query);
        echo "Successfully Deleted";
    }

?>
```