WINTER – 2024 EXAMINATION
Model Answer – Only for the Use of RAC Assessors

Subject Name: Web Based Application development with PHP (WBP) **Subject Code:** 22619

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.
8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a)	**List advantages of PHP.**	**2 M**
	Ans	1)Open Source 2)Platform independent 3)simple and Easy 4)Database support 5)Security 6)Scripting Language 7)Vast Community	Any 4 advantages, 1 for ½ M
	b)	**Define inheritance in PHP.**	**2 M**
	Ans	Inheritance is a mechanism of extending an existing class by inheriting a class. We create a new sub class with all functionality of that existing class, and we	Correct definition 2 M

		can add new members to the new sub class. When we inherit one class from another we say that inherited class is a subclass and the class who has inherits is called parent class. In order to declare that one class inherits the code from another class, we use the extends keyword.	
	c)	**List any four string functions with use.**	**2 M**
	Ans	str_word_count( ): Count the number of words in a string strlen(): Returns the length of a string strrev(): Reverses a string strops(): Returns the position of the first occurrence of a string inside another string (case-sensitive) str_replace(): Replaces some characters in a string (case-sensitive) ucwords(): Convert the first character of each word to uppercase strtoupper(): Converts a string to uppercase letters strtolower(): Converts a string to lowercase letters str_repeat(): Repeating a string with a specific number of times. strcmp(): Compare two strings (case-sensitive). If this function returns 0, the two strings are equal. If this function returns any negative or positive numbers, the two strings are not equal. Substr(): substr() function used to display or extract a string from a particular position. Str_split(): To convert a string to an array Trim(): Removes white spaces and predefined characters from a both the sides of a string. Rtrim(): Removes the white spaces from end of the string Ltrim():Removes the white spaces from left side of the string	List 1 M, Use 1 M, 1/2 M for one function
	d)	**State different data types in PHP.**	**2 M**
	Ans	**i) Integer**: This data type holds only numeric values. Integers hold only whole numbers including positive and negative numbers, i.e., numbers without fractional part or decimal point. The range of integer values are -	Any two data types, 1 M for each

		2,147,483,648 to +2,147,483,647. Integers can be defined indecimal(base 10), hexadecimal(base 16), octal(base 8) or binary(base 2) notation.The default base is decimal (base 10). **(ii) Float**: Floating point numbers represents numeric values with decimal points (real numbers) or a number in exponential form. The range of floating point values are 1.7E-308 to 1.7E+308. Example: 3.14, 0.25, -5.5, 2E-4, 1.2e2 **(iii) String**: A string is a sequence of characters. A string are declares using single quotes or double quotes. Example: "Hello PHP", 'Hello PHP' **(iv) Boolean**: The Boolean data types represents two values, true(1) or false(0). Example: $a=True **v)Array**: An array stores multiple values in one single variable and each value is identify by position (zero is the first position). The array is a collection of heterogeneous (dissimilar) data types. PHP is a loosely typed language that's why we can store any type of values in arrays. Syntax: Variable_name = array (element1, element2, element3, element4.....) **(vi) Object :** Objects are defined as instances of user defined classes that can hold both values and functions. First we declare class of object using keyword class. A class is a structure that contain properties (variables) and methods (functions). **(vii) Resource**: The special resource type is not an actual data type. It is the storing of a reference to functions and resources external to PHP. Consider a function which connect to the database, a function to send a query to the database, a function to close the connection of database. Resource variables hold special handles to opened files, database connections, streams etc. **(viii) Null**: Null is a special data type which can have only one value NULL i.e. a variable that has no value assigned to it. If a variable is created without a value, it is automatically assigned a value of NULL. Syntax : $var_name= NULL	
	e)	**Explain array-flip( ) and explode( ) function ·with syntax.**	**2 M**
	Ans	**array-flip():** The array_flip() function flips/exchanges all keys with their associated values in an array. This built-in function of PHP array_flip() is used to exchange elements within	array-flip(): 1M and explode(): 1M

		an array, i.e., exchange all keys with their associated values in an array and vice-versa. Syntax: array_flip(array); **explode():** - The explode() function breaks a string into an array. - The explode() is a built in function in PHP used to split a string in different strings. - The explode() function splits a string based on a string delimeter, i.e. it splits the string wherever the delimeter character occurs. This function returns an array containing the strings formed by splitting the original string. Syntax: array explode(separator, OriginalString, NoOfElements);	
	f)	**Define class with syntax and example.**	**2 M**
	Ans	- A class is a unit of code that describes the characteristics and behaviors of something, or of a group of things. - Class is a collection of objects. Object has properties and behavior. - Class is a user-defined data type, which includes local functions as well as local data. You can think of a class as a template for making many instances of the same kind (or class) of object. Syntax : <? Php classname_of_class { // code goes here... } ?> Example: <?php class Car	Define:1 M, Syntax: ½ M and any correct Example: ½ M

		{
		// Nothing to see here; move along
		}
		$Maruti = new Car();
		$Honda = new Car();
		print_r($Maruti); // Displays "Car Object ( )"
		print_r($Honda); // Displays "Car Object ( )"
		?>	

	g)	**State advantages of PHP-MySQL.**	**2 M**
	Ans	1. The most important advantage of PHP is that it's open-source and free from cost. It can be downloaded anywhere and is readily available to use for events or web applications. 2. It is platform-independent. PHP-based applications can run on any OS like UNIX, Linux, Windows, etc. 3. Applications can easily be loaded which are based on PHP and connected to the database. It's mainly used due to its faster rate of loading over slow internet speed than other programming language. 4. It has less learning curve because it is simple and straightforward to use. Someone familiar with C programming can easily work on PHP. 5. It is more stable for a few years with the assistance of providing continuous support to various versions. 6. It helps in reusing an equivalent code and not got have to write lengthy code and sophisticated structure for events of web applications. 7. It helps in managing code easily. 8. It has powerful library support to use various function modules for data representation. 9. PHP's built-in database connection modules help in connecting databases easily reducing trouble and time for the development of web applications and content-based sites. 10. The popularity of PHP gave rise to various communities of developers, a fraction of which may be potential candidates for hire. 11. Flexibility makes PHP ready to effectively combine with many other programming languages in order that the software package could use	Any two advantages: 1 M, ½ M for each

		foremost effective technology for every particular feature.	
2.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Explain use of for and for each with example.**	**4 M**

	Ans	**for Statement :**
		
		The for statement is used when you know how many times you want to execute a statement or a block of statements. That is, the number of iterations is known beforehand. These type of loops are also known as entry-controlled loops. There are three main parameters to the code, namely the initialization, the test condition and the counter.
		
		In for loop, a loop variable is used to control the loop. First initialize this loop variable to some value, then check whether this variable is less than or greater than counter value. If statement is true, then loop body is executed and loop variable gets updated. Steps are repeated till exit condition comes.
		
		1. Initialization Expression: In this expression we have to initialize the loop counter to some value. for example: $i=1;
		
		2. Test Expression: In this expression we have to test the condition. If the condition evaluates to true then we will execute the body of loop and go to update expression otherwise we will exit from the for loop. For example: $i<=10;
		
		3.Update Expression: After executing loop body this expression increments/decrements the loop variable by some value. for example : $i+= 2;
		
		Syntax :
		
		for (initialization expression; test condition; update expression)
		
		{
		
		// code to be executed
		
		}
		
		Example:
		
		<?php
		
		//for loop to print even numbers and sum of them
		
		$sum=0;
			for statement: 2 M, for each statement : 2M

		for($i=0; $i<=10;$i+=2)
		{
		echo "$i
		";
		$sum+=$i;
		}
		echo "Sum=$sum";
		?>
		
		for-each statement:
		foreach loop is used for array and objects. For every counter of loop, an array element is assigned and the next counter is shifted to the next element.
		Syntax:
		foreach (array_element as value)
		{
		//code to be executed
		}
		Example:
		<?php
		$arr = array (10, 20, 30, 40, 50);
		foreach ($arr as $i)
		{
		echo "$i
		";
		}
		?>	

	b)	**Demonstrate use of_construct() and_destruct() with example.**	**4 M**
	Ans	**__construct():** To create a constructor, simply add a method with the special name _	__construct():2 M, __destruct(): 2 M

_construct() to your class. (That ' s two underscores, followed by the word " construct, " followed by parentheses.) PHP looks for this special method name when the object is created; if it finds it, it calls the method.

Example :

```
class MyClass

{

function __construct()

{

echo "Welcome to PHP constructor. <br / > ";

}

}

$obj = new MyClass; // Displays "Welcome to PHP constructor."
```

__ destruct(): A destructor is called when the object is destroyed.

- You have to manually dispose of objects you created, but in PHP, it's handled by the Garbage Collector, which keeps an eye on your objects and automatically destroys them when they are no longer needed.

- Destructors are useful for tidying up an object before it's removed from memory.

- Destructors don't have any types or return value. It is just called before de-allocating memory for an object or during the finish of execution of PHP scripts or as soon as the execution control leaves the block.

- For example, if an object has a few files open or contains data that should be written to a database, it's a good idea to close the files or write the data before the object disappears.

- You can create destructor methods in the same way as constructors, except that you use __destruct() function

Example:

```
<?php

class MyDestClass
```

		{
		function __construct()
		{
		print "In constructor
		";
		}
		function __destruct()
		{
		print "Destroying " . __CLASS__ . "
		";
		}
		}
		$obj = new MyDestClass();//object is not created
		?>	

	c)	**Write PHP program** **i) To find largest of two number** **ii) for connecting to MySQL server.**	**4 M**

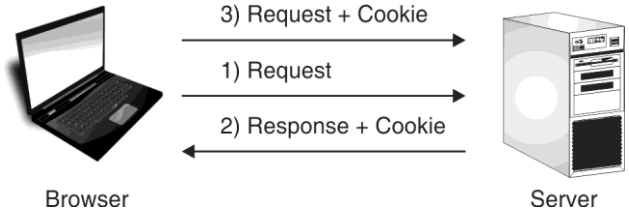
	Ans	<?php
		
		function findLargest($num1, $num2) {
		if ($num1 > $num2) {
		return $num1;
		} elseif ($num2 > $num1) {
		return $num2;
		} else {
		return "Both numbers are equal";
		}
		}
		
		$num1 = 10;
		$num2 = 20;
		
		echo "The largest number between $num1 and $num2 is: " . findLargest($num1, $num2);
			Any Correct logic Program for i) 2 M ii) 2 M

		?>
		
		ii) <?php
		// MySQL server credentials
		$servername = "localhost"; // Change this if your MySQL server is on a different host
		$username = "root"; // MySQL username (default: root)
		$password = ""; // MySQL password (default: empty string for local MySQL setup)
		$dbname = "your_database"; // Name of the database you want to connect to (change as necessary)
		// Create connection using MySQLi
		$conn = new mysqli($servername, $username, $password, $dbname);
		// Check the connection
		if ($conn->connect_error) {
		die("Connection failed: " . $conn->connect_error);
		}
		echo "Connected successfully to the MySQL server";
		// Close the connection
		$conn->close();
		?>	

	d)	**Write a PHP program to count total number of rows in the database table.**	**4 M**

	Ans	<?php
		// MySQL server credentials
		$servername = "localhost"; // Change this if your MySQL server is on a different host
		$username = "root"; // MySQL username (default: root)
			Any correct logic program 4 M

```
$password = "";            // MySQL password (default: empty string for local MySQL setup)

$dbname = "your_database"; // Name of the database you want to connect to

// Create connection using MySQLi

$conn = new mysqli($servername, $username, $password, $dbname);

// Check the connection

if ($conn->connect_error) {

   die("Connection failed: " . $conn->connect_error);

}

// Define the table name

$tableName = "your_table_name"; // Replace this with your actual table name

// SQL query to count total number of rows in the table

$sql = "SELECT COUNT(*) AS total_rows FROM $tableName";

// Execute the query

$result = $conn->query($sql);

// Check if the query was successful

if ($result->num_rows > 0) {

   // Fetch the result as an associative array

   $row = $result->fetch_assoc();

      // Display the total number of rows

   echo "Total number of rows in the '$tableName' table: " . $row['total_rows'];

} else {

   echo "Error: Could not retrieve row count.";

}

// Close the connection

$conn->close();
```

		?>	
3.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Define cookie. Explain how to create and delete cookies.**	**4 M**
	Ans	– 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. In short, cookie can be created, sent and received at server end. – A cookie is a small file with the maximum size of 4KB that the web server stores on the client computer. – A cookie can only be read from the domain that it has been issued from. Cookies are usually set in an HTTP header but JavaScript can also set a cookie directly on a browser.  There are three steps involved in identifying returning users : 1. Server script sends a set of cookies to the browser. For example : name, age or identification number etc. 2. Browser stores this information on local machine for future use. 3. When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user. – PHP provides a inbuilt function setcookie(), that can send appropriate HTTP header to create the cookie on the browser. – While creating the cookie, we have to pass require arguments with it. – Only name argument is must but it is better to pass value, expires and path to avoid any ambiguity. **Syntax :** setcookie(*name, value, expire, path, domain, secure, HttpOnly*); setcookie(“username”, “abc”, time() + 60 * 60 * 24 * 365, “/”, “.abc.com”, false, true); – The setcookie() function defines a cookie to be sent along with the rest of the HTTP headers. – cookieexample.php <html> <body>	Definition of cookie-1M Creation of cookie with syntax and example-1.5M Deletion of cookie with syntax and example-1.5M

```
<?php
$cookie_name = "username";
$cookie_value = "abc";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400
= 1 day

if(!isset($_COOKIE[$cookie_name])) {
   echo "Cookie name '" . $cookie_name . "' is not set!";
} else {
   echo "Cookie '" . $cookie_name . "' is set!<br>";
   echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
</body>
</html>
```

Delete Cookies

- Cookie can be deleted from user browser simply by setting expires argument to any past date it will automatically delete the cookie from user browser.
- Deleted cookie can be checked by calling the same cookie with its name to check if it exists or not.
- There is no special dedicated function provided in PHP to delete a cookie. All we have to do is to update the expire-time value of the cookie by setting it to a past time using the setcookie() function. A very simple way of doing this is to deduct a few seconds from the current time.
- Syntax:
- setcookie(name, time() - 3600);

```
<html>
<body>
<?php
setcookie("user"," ",time()-3600);
echo "Cookie 'user' is deleted.";
?>
</body>
</html>
```

	b)	**Explain bitwise operators in PHP.**	**4 M**

Ans Bitwise operators allow evaluation and manipulation of specific bits within an integer.

Bitwise Operators		
Example	**Name**	**Result**
$a & $b	AND	Bits that are set in both *$a* and *$b* are set.
$a | $b	OR (inclusive OR)	Bits that are set in either *$a* or *$b* are set.
$a ^ $b	Xor (exclusive or)	Bits that are set in *$a* or *$b* but not both are set.
~ $a	Not	Bits that are set in *$a* are not set, and vice versa.
$a << $b	Shift left	Shift the bits of *$a $b* steps to the left (each step means "multiply by two")
$a >> $b	Shift right	Shift the bits of *$a $b* steps to the right (each step means "divide by two")

1. & (Bitwise AND) : This binary operator works on two operands. Bitwise AND operator in PHP takes two numbers as operands and does AND on every bit of two numbers. The result of AND is 1 only if both bits are 1.

2. | (Bitwise OR) :Bitwise OR operator takes two numbers as operands and does OR on every bit of two numbers. The result of OR is 1 any of the two bits is 1.

3. ^ (Bitwise XOR) : This is also known as Exclusive OR operator. Bitwise XOR takes two numbers as operands and does XOR on every bit of two numbers. The result of XOR is 1 if the two bits are different.

4. ~ (Bitwise NOT) : This is a unary operator i.e. works on only one operand. Bitwise NOT operator takes one number and inverts all bits of it.

5. << (Bitwise Left Shift) :Bitwise Left Shift operator takes two numbers, left shifts the bits of the first operand, the second operand decides the number of places to shift.

6. >> (Bitwise Right Shift) :Bitwise Right Shift operator takes two numbers, right shifts the bits of the first operand, the second operand decides the number of places to shift.

Example:

```
// Bitwise AND

   $First = 5;
```

List of Bitwise operators with use-2M

And example of any bitwise operator-2M

```
$second = 3;
$answer = $First & $second;
print_r("Bitwise & of 5 and 3 is $answer");
print_r("\n");
// Bitwise OR
$answer = $First | $second;
print_r("Bitwise | of 5 and 3 is $answer");
print_r("\n");
// Bitwise XOR
$answer = $First ^ $second;
print_r("Bitwise ^ of 5 and 3 is $answer");
print_r("\n");
// Bitwise NOT
$answer = ~$First;
print_r("Bitwise ~ of 5 is $answer");
print_r("\n");
// Bitwise Left shift
$second = 1;
$answer = $First << $second;
print_r("5 << 1 will be $answer");
print_r("\n");
// Bitwise Right shift
$answer = $First >> $second;
print_r("5 >> 1 will be $answer");
print_r("\n");
?>
```

Output:

```
Bitwise & of 5 and 3 is 1
Bitwise | of 5 and 3 is 7
```

		Bitwise ^ of 5 and 3 is 6 Bitwise ~ of 5 is -6 5 << 1 will be 10 5 >> 1 will be 2	
	c)	**Write a PHP program to draw a rectangle filled with red colour.**	**4 M**

	Ans	<?php
		// Create an image
		$img = imagecreatetruecolor(500, 300);
		$color = imagecolorallocate($img, 255, 0, 0);
		imagefilledrectangle($img, 30, 30, 470, 270, $color);
		header("Content-type: image/png");
		imagepng($img);
		imagedestroy($img);
		?>
			Relevant program-4M

	d)	**Explain Following Methods** **i) MySQL_select_db()** **ii) ImageCopyReszied()**	**4 M**

	Ans	**i) MySQL_select _db()**
		The mysqli_select_db() function is used to change the default database for the connection.
		Syntax: $mysqli -> select_db(*name*)
		Example:
		$con=mysqli_connect("localhost","my_user","my_password","my_db");
		if (mysqli_connect_errno()) {
		echo "Failed to connect to MySQL: " . mysqli_connect_error();
		exit;
		}
		// Return name of current default database
		if ($result = mysqli_query($con, "SELECT DATABASE()")) {
		$row = mysqli_fetch_row($result);
		echo "Default database is " . $row[0];
		mysqli_free_result($result);
			MySQL_select_db()-use and syntax:1M
			Example-1M
			
			ImageCopyResized()-use and syntax:1M
			Example-1M

```
}

// Change db to "test" db
mysqli_select_db($con, "test");

// Return name of current default database
if ($result = mysqli_query($con, "SELECT DATABASE()")) {
  $row = mysqli_fetch_row($result);
  echo "Default database is " . $row[0];
  mysqli_free_result($result);
}

// Close connection
mysqli_close($con);
?>
```

ii) imageCopyReszied()

The **imagecopyresized() function** is an inbuilt function in PHP which is used to copy a rectangular portion of one image to another image. dst_image is the destination image, src_image is the source image identifier.

Syntax:

bool imagecopyresized(*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)

Parameters: This function accepts ten parameters as mentioned above and described below:

- **$dst_image:** It specifies the destination image resource.
- **$src_image:** It specifies the source image resource.
- **$dst_x:** It specifies the x-coordinate of destination point.
- **$dst_y:** It specifies the y-coordinate of destination point.
- **$src_x:** It specifies the x-coordinate of source point.
- **$src_y:** It specifies the y-coordinate of source point.
- **$dst_w:** It specifies the destination width.
- **$dst_h:** It specifies the destination height.
- **$src_w:** It specifies the source width.

		• **$src_h:** It specifies the source height.
		
		Return Value: This function returns TRUE on success or FALSE on failure.
		
		<?php
		// File and new size
		$filename = 'test.jpg';
		$percent = 0.5;
		
		// Content type
		header('Content-Type: image/jpeg');
		
		// Get new sizes
		list($width, $height) = getimagesize($filename);
		$newwidth = $width * $percent;
		$newheight = $height * $percent;
		
		// Load
		$thumb = imagecreatetruecolor($newwidth, $newheight);
		$source = imagecreatefromjpeg($filename);
		
		// Resize
		imagecopyresized($thumb, $source, 0, 0, 0, 0, $newwidth, $newheight, $width, $height);
		
		// Output
		imagejpeg($thumb);
		?>	

4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Demonstrate the concept of overloading with example.**	**4 M**
	Ans	It is a type of overloading for creating dynamic methods that are not declared within the class scope. PHP method overloading also triggers magic methods dedicated to the appropriate purpose. Unlike property overloading, PHP method overloading allows function call on both object and static context. The related magic functions are, • __call() – triggered while invoking overloaded methods in the object context. • __callStatic() – triggered while invoking overloaded methods in static context. <?php	**concept of overloading- 2M and example- 2M**

```
class GFG {

    public function __call($name, $arguments) {



    echo "Calling object method '$name' "

      . implode(', ', $arguments). "\n";

  }

  public static function __callStatic($name, $arguments)
{
        echo "Calling static method '$name' "
      .implode(', ', $arguments). "\n";
  }
}
 // Create new object
$obj = new GFG;
 $obj->runTest('in object context');
 GFG::runTest('in static context');
 ?>

<?php

class GFG {
   function multiply($var1){
      return $var1;
   }


   function multiply($var1,$var2){
      return $var1 * $var1 ;
   }
}
  $ob = new GFG();
$ob->multiply(3,6);
?>
```

b) **Develop PHP program to create database and insert records in database.** **4 M**

Ans

Create table in php:

Step 1) create a database through script

```
<?php
if(isset($_POST)) {
   $servername = "localhost";
   $username = "root";
```

create database -2M

and insert records in database-2M

```
   $password = "";
   //$dbname = "clg";
   $conn = new mysqli($servername, $username, $password);
   if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
   }

         $sql = "CREATE DATABASE clg";
if ($conn->query($sql) === TRUE) {
  echo "Database created successfully";
} else {
  echo "Error creating database: " . $conn->error;
}
$conn->close();
}
?>
```

) Create a table “staff”

```
<?php
 {
   $servername = "localhost";
   $username = "root";
   $password = "";
   $dbname = "clg";
   $conn = new mysqli($servername, $username, $password, $dbname);
   if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
   }
   $sql = "CREATE TABLE staff (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50)
)";
if ($conn->query($sql) === TRUE) {
  echo "Table staff created successfully";
} else {
  echo "Error creating table: " . $conn->error;
}
   $conn->close();
}

?>
```

insert values into table “staff”

```
<?php
 {
   $servername = "localhost";
```

```
    $username = "root";
    $password = "";
    $dbname = "clg";
    $conn = new mysqli($servername, $username, $password, $dbname);
    if ($conn->connect_error) {
       die("Connection failed: " . $conn->connect_error);
    }
    $sql = "INSERT INTO staff (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if ($conn->query($sql) === TRUE) {
  echo "New record created successfully";
} else {
  echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
}
?>
```

	c)	**Explain how to implement multidimensional array in PHP.**	**4 M**
	Ans	These are arrays that contain other nested arrays. An array which contains single or multiple arrays within it and can be accessed via multiple indices. We can create one dimensional and two dimensional array using multidimensional arrays. The advantage of multidimensional arrays is that they allow us to group related data together. Syntax for creating multidimensional array in php array (array (elements...), array (elements...), ...) Example : <?php // Defining a multidimensional array $person = array(array("name" => "Yogita K", "mob" => "5689741523", "email" => "yogi_k@gmail.com",), array("name" => "Manisha P.", "mob" => "2584369721", "email" => "manisha_p@gmail.com",), array("name" => "Vijay Patil", "mob" => "9875147536",	Concept and syntax of multidimensional array- 2M Implementation/example- 2M

		"email" => "Vijay_p@gmail.com",)); // Accessing elements echo "manisha P's email-id is: " . $person[1]["email"], " "; echo "Vijay Patil's mobile no: " . $person[2]["mob"]; ?> Output : manisha P's email-id is: manisha_p@gmail.com Vijay Patil's mobile no: 9875147536	
	d)	**Explain print and echo statement in PHP.**	**4 M**

	Ans	PHP echo statement:
		• In PHP, there are two ways to get output or print output: echo and print.
		• They are both used to output data to the screen.
		The PHP echo Statement:
		• The echo statement can be used with or without parentheses: echo or echo().
		• The echo statement can display anything that can be displayed to the browser, such as string, numbers, variables values, the results of expressions etc.
		Example :
		<?php
		// Displaying strings
		echo "Hello, Welcome to PHP Programming";
		echo "
		";
		//Displaying Strings as Multiple arguments
		echo "Hello", " Welcome", " PHP";
		echo "
		";
		//Displaying variables
		$s="Hello, PHP";
		$x=10;
		$y=20;
		echo "$s
		";
		echo $x."+".$y."=";
			Use of print and echo statement-2M
			Example of each-2M

echo $x + $y;

?>

Output :

Hello, Welcome to PHP Programming

Hello Welcome PHP

Hello, PHP

10+20=30

The PHP print Statement :

The PHP print statement is similar to the echo statement and can be used alternative to echo at many times. It is

also language construct and so we may not use parenthesis : print or print(). print statement can also be used to print

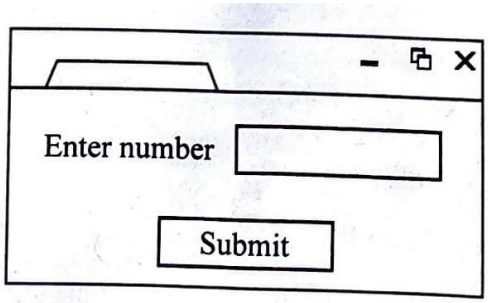
strings and variables.

Web Based Application Development using PHP (MSBTE) 1-6 Expressions and Control Statements in PHP

Example :

```
<?php
// Displaying strings
print "Hello, Welcome to PHP Programming";
print "<br/>";
//Displaying variables
$s="Hello, PHP";
$x=10;
$y=20;
print "$s <br/>";
print $x."+".$y."=";
print $x + $y;
?>
```

Output :

		Hello, Welcome to PHP Programming Hello, PHP 10+20=30	
	e)	**Create a form given below also write PHP code to find given number is odd or even.**  **Fig. – 1**	**4 M**

	Ans	<?php
		extract($_REQUEST);
		if(isset($check))
		{
		if($number%2==0)
		{
		echo "$number is Even Number";
		}
		else
		{
		echo "$number is Odd Number";
		}
		}
		?>
		<!DOCTYPE html>
		<html>
		<head>
		<title>Form</title>
		</head>
		<body>
		<form method="post">
		<table>
		<tr>
		<td>Enter Your Number</td>
		<td><input type="text" name="number" required/></td>
		</tr>
		<td colspan="2" align="center">
		<input type="submit" value="Even or Odd" name="check"/>
			Relevant code-4M

```
                </tr>
                </form>
        </body>
</html>
</html>
```

OR

```
<html>
<body>

   <h2>PHP script to find given number is ODD or EVEN </h2>

   <form action="" method="post">
      <input type="text" name="num" />
      <input type="submit" />
   </form>

   <?php

      if($_POST)
      {
         $num = $_POST['num'];

         if(!is_numeric($num))
         {
            echo "String not allowed.
            Input should number";
            return;
         }
         if($num % 2==0)
         {
            echo "Number is an even number";
         }
         else
         {
            echo "Number is an odd number ";
         }
      }

   ?>

</body>
</html>
```

5.		**Attempt any <u>TWO</u> of the following:**	**12 M**

	a)	**State applications of serialization. Illustrate its use with example.**	**6 M**
	Ans	Serialization in PHP refers to the process of converting a data structure (like an object, array, or variable) into a string format that can be stored, transmitted, or manipulated. Deserialization is the reverse process, converting the serialized string back into its original data structure. **Applications of serialization in PHP:** 1. Serialization allows developers to store complex data structures, like objects and arrays, in databases or files. 2. When data needs to be sent over a network (e.g., via APIs or sockets), it is serialized to ensure compatibility with the transmission format. 3. PHP sessions often use serialization to store session data as strings. The session data is serialized before being written to a storage medium like files or databases. 4. Serialization is used in caching systems like Redis or Memcached to store PHP objects or arrays for faster retrieval. 5. Serialization can preserve the state of objects by storing their properties and values. This is useful when saving the state of an object between requests or sessions. 6. Developers use serialization to log complex data structures in a readable and storable format for debugging purposes. 7. Serialization is often used in queue systems to serialize tasks or messages before pushing them into the queue. 8. When applications written in PHP need to communicate with systems in other programming languages, serialization ensures that the data can be properly transmitted and interpreted. Example: <?php class User { public $name; public $email; public function **__construct**($name, $email) {	Any three correct applications-3 M Any Correct example in PHP – 3 M

```
        $this->name = $name;
        $this->email = $email;
    }
}
$user = new User('Rina', 'rina@gmail.com');
$serializedUser = serialize($user);
echo $serializedUser;
?>
```

OUTPUT:

```
O:4:"User":2:{s:4:"name";s:4:"Rina";s:5:"email";s:15:"rina@gmail.com";}
```

b) Write a PHP program

(i) for sending mail

(ii) for validating name field. — **6 M**

Ans

(i) PHP program to sending mail

```
<html>
<head>
<title>Email using PHP</title>
</head>
<body>
<?php
$to_email = "spat20.06.86@gmail.com";
$subject = 'Testing PHP Mail';
$message = 'This mail is sent using the PHP mail function';
$headers = 'From: spat20.06.86@gmail.com';
$retvalue= mail($to_email,$subject,$message,$headers);
echo $retvalue;
if( $retvalue == true ) {
        echo "Message sent successfully...";
}else {
        echo "Message could not be sent...";
      }
?>
</body>
</html>
```

(ii) For Validating name field

```
<!DOCTYPE html>
<html>
<head>
```

for sending mail-3M
for validating name field-3M

```
<style>
.error {color: #FF0001;}
</style>
</head>
<body>
<?php
// define variables to empty values
$nameErr = " ";
//Input fields validation
if ($_SERVER["REQUEST_METHOD"] == "POST") {
// Validating Name field
   if (emptyempty($_POST["name"])) {
       $nameErr = "Name is required";
   } else {
      $name = input_data($_POST["name"]);
         // check if name only contains letters and whitespace
         if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
            $nameErr = "Only alphabets and white space are allowed";
         }
   }
?>
<h2>Input Form</h2>
<span class = "error">* required field </span>
<br><br>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>" >
  Name:  <input type="text" name="name">
  <span class="error">* <?php echo $nameErr; ?> </span>
    <br><br>
<input type="submit" name="submit" value="Submit">
    <br><br>
</form>
<?php
    if(isset($_POST['submit'])) {
    if($nameErr == "") {
       echo "<h3 color = #FF0001> <b>You have sucessfully
registered.</b> </h3>";
       echo "<h2>Your Input:</h2>";
       echo "Name: " .$name;
       echo "<br>";
     } else {
       echo "<h3> <b>Please input name in correct fomat.</b> </h3>";
    }
    }
?>
</body>
</html>
```

	c)	**Display the given text "this is server side coding using PHP" in**	**6 M**

		PDF format using PHP.	
	Ans	<?php require('fpdf.php'); $pdf=new FPDF(); $pdf->SetFillColor(100,256,256); $pdf->AddPage(); $pdf->SetFont('Courier','B',16); $pdf->SetTextColor(0,100,200); $pdf->Cell(100,10,this is server side coding using PHP',0,1,'C',true); $pdf->Output(); ?>	Correct code-6M
6.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Create a html form "result.html" to accept Rollno of student using submit button. Write "result.php" code to check the result of student "pass" or "fail". Create a table result_table in MySQL database "My_db" with two columns Rollno and Status. Also write PHP code to delete a record from result_table.**	**6 M**
	Ans	**File result.html** <html> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Check Result</title> </head> <body> Check Student Result <form action="result.php" method="POST"> <label for="rollno">Enter Roll Number:</label>	Correct Result.html-2 M Result.php -2 M Create table / Insert two records/ Delete the record- 2 M

```
        <input type="number" name="rollno" id="rollno" required>
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

PHP Code: result.php

```
<?php
// Database credentials
$servername = "localhost";
$username = "root";
$password = ""; // Adjust if needed
$dbname = "My_db";
// Connect to the database
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Get the roll number from the form
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $rollno = intval($_POST['rollno']);
    // Query to check the result
    $sql = "SELECT Status FROM result_table WHERE Rollno = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("i", $rollno);
    $stmt->execute();
    $result = $stmt->get_result();
    // Check if the Rollno exists
```

```
    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        echo "Roll Number: " . $rollno . "<br>";
        echo "Result: " . $row['Status'];
    } else {
        echo "No result found for Roll Number: " . $rollno;
    }
   $stmt->close();
}
$conn->close();
?>
```

Create table result_table

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = " My_db";
// Creating connection
$conn = new mysqli($servername, $username, $password,$dbname);
// Checking connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully<br>";
// sql to create table
$sql = "CREATE TABLE result_table ( Rollno INT PRIMARY KEY, Status
VARCHAR(10) NOT NULL)";
if ($conn->query($sql) === TRUE) {
    echo "Table created successfully";
```

```
} else {
   echo "Error creating table: " . $conn->error;
}
$conn->close();
?>
```

File - Insert.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "My_db";
// Creating connection
$conn = new mysqli($servername, $username, $password,$dbname);
// Checking connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully<br>";
$sql = "INSERT INTO result_table (Rollno, Status)
VALUES (1, 'Pass');";
$sql .= " INSERT INTO result_table (Rollno, Status)
VALUES (2, 'Fail')";
if ($conn->multi_query($sql) === TRUE) {
   echo "New records created successfully";
} else {
   echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
```

		?>
		Code to delete the record:
		<?php
		$servername = "localhost";
		$username = "root";
		$password = "";
		$dbname = "My_db";
		// Creating connection
		$conn = new mysqli($servername, $username, $password,$dbname);
		// Checking connection
		if ($conn->connect_error) {
		die("Connection failed: " . $conn->connect_error);
		}
		echo "Connected successfully
		";
		$sql = "DELETE from result_table WHERE Rollno=2";
		if ($conn->query($sql) === TRUE) {
		echo "record deleted successfully";
		} else {
		echo "Error: " . $sql . "
		" . $conn->error;
		}
		$conn->close();
		?>	

	b)	**Explain the following function types with example:** **(i) Anonymous function** **(ii) Variable function**	**6 M**
	Ans	(i) **Anonymous function or lambda function**: Anonymous function is a function without any user defined name. Such a function is also	Anonymous function with example – 3 M, Variable function with example – 3 M

called closure or lambda function.

Syntax:

$var=function ($arg1, $arg2) { return $val; };

• There is no function name between the function keyword and the opening

parenthesis.

• There is a semicolon after the function definition because anonymous function

definitions are expressions.

• Function is assigned to a variable, and called later using the variable's name.

• When passed to another function that can then call it later, it is known as a callback.

• Closure is also an anonymous function that can access variables outside its scope

with the help of use keyword

Example-1 :

```
<?php
$var = function ($name)
{
echo "Hello $name";
};
$var("Sneha");
?>
```

OUTPUT:

Hello Sneha

Example 2 - Anonymous function as a Closure:

```
<?php
$maxmarks=300;
$percent=function ($marks) use ($maxmarks)
{
```

```
 return $marks*100/$maxmarks;
};
echo "marks=285 percentage=". $percent(285);
?>
```

(ii) Variable function:

If name of a variable has parentheses (with or without parameters in it) in front of it, PHP

parser tries to find a function whose name corresponds to value of the variable and executes

it. Such a function is called variable function.

Syntax:

```
<?php
 function valueofvariable (arg list)
 {
 //block of code;
 }
 $variable_name= valueofvariable;
 $variable_name (arg list);
?>
```

Example:

```
<?php
function add($x, $y){
 echo $x+$y;
}
function sub($x, $y){
 echo $x-$y;
}
$var=readline("enter name of function: ");
$var(10,20);
```

		?> **OUTPUT** add 30	
	c)	**Explain:** **(i) _clone()** **(ii) class_exists()** **(iii) get_parent_class()**	**6 M**

	Ans	**(i) _clone()**
		By combining the clone keyword and __clone() magic method, we can perform a deep copy of an object. Deep copy creates a copy of an object and recursively creates a copy of the objects referenced by the properties of the object.
		The following example illustrates how to use the __clone() magic method to carry a deep copy of the Person object:
		Example:
		<?php
		class Address
		{
		public $street;
		public $city;
		}
		class Person
		{
		public $name;
		public $address;
		public function __construct($name)
		{
		$this->name = $name;
		$this->address = new Address();
			Correct explanation Of _clone()-2 M
			class_exists()-2 M
			get_parent_class()-2 M

```
}

public function __clone()

{

$this->address = clone $this->address;

}

}

$bob = new Person('Bob');

$bob->address->street = 'North 1st Street';

$bob->address->city = 'San Jose';

$alex = clone $bob;

$alex->name = 'Alex';

$alex->address->street = '1 Apple Park Way';

$alex->address->city = 'Cupertino';

var_dump($bob);

var_dump($alex);

?>
```

In above example, using __clone() method in the Person class we can create a separate

copy of the Address object for both the references which is known as deep copy of

object,.

(ii) class_exists()

The class_exists() function checks whether a class with a given name has been defined or not. It is useful for verifying the availability of a class before instantiating it or calling its methods.

Syntax

```
bool class_exists(string $className, bool $autoload = true)
```

Example:

```
<?php
```

```
class MyClass {}

if (class_exists('MyClass')) {

   echo "Class 'MyClass' exists.";

} else {

   echo "Class 'MyClass' does not exist.";

}

?>
```

Output:

Class 'MyClass' exists.

(iii) get_parent_class()

The get_parent_class() function retrieves the name of the parent class for a given class or object. It is used in object-oriented programming to examine class inheritance.

Syntax

```
string|false get_parent_class(object|string $objectOrClass = null)
```

Example:

```
<?php

class ParentClass {}

class ChildClass extends ParentClass {}

$child = new ChildClass();

echo get_parent_class($child);

?>
```

Output:

ParentClass