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

Subject Name: Web Based Application Development with PHP **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 any four advantages of PHP.**	**2 M**
	Ans	1. **Performance:** PHP script is executed much faster than those scripts which are written in other languages such as JSP and ASP. PHP uses its own memory, so the server workload and loading time is automatically reduced, which results in faster processing speed and better performance. 2. **Open Source:** PHP source code and software are freely available on the web. We can develop all the versions of PHP according to our requirement without paying any cost. All its components are free to download and use. 3. **Embedded:** PHP code can be easily embedded within HTML tags and script. 4. **Platform Independent:** PHP is available for WINDOWS, MAC, and LINUX& UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also. 5. **Database Support:** PHP supports all the leading databases such as MySQL, SQLite, ODBC, etc. 6. **Security:** PHP is a secure language to develop the website. It consists of multiple	Any 4 correct points-2 M

		layers of security to prevent threads and malicious attacks. 7. **Loosely Typed Language:** PHP allows us to use a variable without declaring its datatype. It will be taken automatically at the time of execution based on the type of data it contains on its value. 8. **Error Reporting:** PHP has predefined error reporting constants to generate an error notice or warning at runtime. E.g., E_ERROR, E_WARNING, E_STRICT, E_PARSE. 9. **Control:** Different programming languages require long script or code, whereas PHP can do the same work in a few lines of code. It has maximum control over the websites like we can make changes easily whenever we want. 10. **Web servers Support:** PHP is compatible with almost all local servers used today like Apache, Netscape, Microsoft IIS, etc. 11. **Familiarity with syntax:** PHP has easily understandable syntax. Programmers are comfortable coding with it. 12. **Helpful PHP Community:** It has a large community of developers who regularly updates documentation, tutorials, online help, and FAQs. Learning PHP from the communities is one of the significant benefits.	
	b)	**State the use of str-word-count along with its syntax.**	**2 M**
	Ans	**Use of str_word_count() :** The str_word_count() is in-built function of PHP. It is used to return information about words used in a string or counts the number of words in a string. **Syntax:** str_word_count ($string , $returnVal, $chars) **Parameters Used:** $string: Input string. $returnVal: $returnVal parameter. It is an optional parameter and its default value is 0. $chars: This is an optional parameter which specifies a list of additional characters which shall be considered as a 'word'.	Use -1 M and Correct Syntax- 1 M
	c)	**Define Serialization.**	**2 M**
	Ans	In PHP, serialization is the process of converting a data structure or object into a string representation that can be stored or transmitted. This allows you to save the state of an object or data structure to a database, cache, or send it over a network connection, and then later recreate the object or data structure from the serialized string.	Correct Definition- 2 M

d) Differentiate between Session & Cookies. (Any two points) — **2 M**

Ans — Any 2 correct points – 2 M

Sr. No.	Session	Cookies
1	When it comes to sessions, it stores the variable and their values within a file in a momentary manual on the server.	When it comes to cookies, they are stored as a text file on the user's system.
2	It ends when the operator turns off the system or logs off the application.	The cookie expires relying on the lifetime a user set for it.
3	They are server-side files that manage user data.	They are considered client-side files that store user data.
4	It can hold an unlimited amount of data.	It can only hold a specific piece of information.
5	Here, we can hold unlimited data; however, there is the highest memory limitation, which a script can utilise a single time, and it is 128 MB.	Here, the utmost size of the cookies is 4 KB.
6	To begin the session, we are required to call the session_start() function.	As cookies are stored in the local system hence, we don't require to call a function to initiate a cookie.
7	Sessions are more secure than cookies.	Cookies are less secure as compared to the sessions.

e) List out the database operations. — **2 M**

Ans — Any 4 correct operations-2 M

1. Create
2. Insert
3. Update
4. Delete
5. Truncate
6. Alter

f) Write a program using Foreach loop. — **2 M**

Ans — Correct program- 2 M

```
<?php
   //declare array
   $season = array ("Summer", "Winter", "Autumn", "Rainy");

   //access array elements using foreach loop
   foreach ($season as $element) {
```

		echo "$element"; echo "</br>"; } ?>	
	g)	**Explain in short how can we destroy cookies.**	**2 M**

	Ans	To delete or destroy the cookie, we can use the setcookie() function with the same cookie name, an empty value, and a past expiration time (specifically, one hour ago). Setting the expiration time to the past causes the cookie to be immediately expired and removed from the browser.
		Example:
		<?php
		$cookieName = "username";
		
		// Set the cookie expiration time to the past to delete the cookie
		setcookie($cookieName, "", time() - 3600);
		
		echo "Cookie named 'username' has been deleted.";
		?>
			Correct explanation-2 M

2.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Explain different loops in PHP.**	**4 M**

	Ans	**Type of Loops in PHP**
		1. The while Loop
		2. The Do …while Loop
		3. The For Loop
		4. The Foreach Loop
		
		The While Loop
		While structure is type of loop statement, where the condition is checked at first, the iteration will not stop even if the value changes while executing Statements.
		Syntax-
		while(condition)
		{
		code to be executed
		}
		Example:
		<?php
		$i=0;
		while($i<=10)
		{ //output value 0 from 10
		echo "The Number is ".$i."
		";
			Types of Loops-1 M
			Any 2 correct Loop with syntax-2 M
			Example of above loop-1 M

```
$i++;
}
?>
```

The Do While Loop

Do while statement is same as the while statement, the only difference is that it evaluates the expression at the end.

Syntax –

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

Example:

```
<?php
$i=0;
do
{ //output value 0 from 10
echo "The Number is ".$i."<br>";
$i++;
}
while($i<=10)
?>
```

The For Loop

The for loop is used when we know in advance how many times the script should run.

Syntax:

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

Example:

```
<?php
for($i=0;$i<=10;$i++)
{
echo "The Number is ".$i."<br/>";
}
?>
```

The Foreach Loop

For Each structure is a loop structure used for arrays.

Syntax:

```
foreach(array as value)
{
code to be executed;
}
foreach(array as key => value)
{
```

code to be executed;
}

Example:

```
<?php
$email=array('info@phpgurukul.com','anuj.lpu1@gmail.com');
foreach($email as $value)
{
echo "Processing ".$value."<br/>";
}
?>
```

b) Differentiate between implode & explode functions. **4 M**

Ans

Implode function	Explode function
The implode function works on an array.	The explode function works on a string.
The implode function returns string.	The explode function returns array.
The first parameter of the implode function is optional.	The first parameter of the explode function is required.
The implode function in PHP is used to "join elements of an array with a string".	The explode function is used to "Split a string by a specified string into pieces i.e. it breaks a string into an array".
Syntax: implode (separator, array)	Syntax: explode (separator,string,limit)

Any 4 Correct points- 4 M

c) Define Introspection and explain it with suitable example. **4 M**

Ans

Definition:

In PHP, An Introspection is an ability to examine an object's characteristics, such as its name, parent class (if any) properties, classes, interfaces, and methods.

Definition-1 M

In-built functions in PHP Introspection:

1. **class_exists()** Checks whether a class has been defined.
2. **get_class()** Returns the class name of an object.
3. **get_parent_class()** Returns the class name of a Return object's parent class.

Any 4 correct Introspection built-in functions – 1 M

Any Correct Example- 2

4. **is_subclass_of()** Checks whether an object has a given parent class.
5. **get_declared_classes()** Returns a list of all declared classes.
6. **get_class_methods()** Returns the names of the class methods.
7. **get_class_vars()** Returns the default properties of a class
8. **interface_exists()** Checks whether the interface is defined.
9. **method_exists()** Checks whether an object defines a method.

M

Example:

```
<?php
class Rectangle
{
 var $dim1 = 2;
 var $dim2 = 10;
function Rectangle($dim1,$dim2)
{
 $this->dim1 = $dim1;
 $this->dim2 = $dim2;
}
function area()
{
 return $this->dim1*$this->dim2;
}
function display()
{
 // any code to display info
}
}

$S = new Rectangle(4,2);
//get the class varibale i.e properties
$class_properties = get_class_vars("Rectangle");
//get object properties
$object_properties = get_object_vars($S);
//get class methods
$class_methods = get_class_methods("Rectangle");
//get class corresponding to an object
$object_class = get_class($S);
print_r($class_properties);
print_r($object_properties);
print_r($class_methods);
print_r($object_class);
?>
```

OUTPUT:

```
Array ( [dim1] => 2 [dim2] => 10 )
Array ( [dim1] => 4 [dim2] => 2 )
```

		Array ([0] => Rectangle [1] => area [2] => display) Rectangle	
	d)	**Explain mail () function in PHP with example.**	**4 M**
	Ans	PHP mail() function is used to send email in PHP. You can send text message, html message and attachment with message using PHP mail() function. Syntax: bool mail (string $to , string $subject , string $message [, string $additional_headers [, string $additional_parameters]]) Description: $to: specifies receiver or receivers of the mail. $subject: represents subject of the mail. $message: represents message of the mail to be sent. $additional_headers (optional): specifies the additional headers such as From, CC, BCC etc. Example: <html> <head> <title>Email Using PHP</title> </head> <body> <?php $to_email = "sneha.patange@vpt.edu.in"; $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) {	Syntax with description-2 M Example-2 M

```
        echo "Message sent successfully...";
}else {
        echo "Message could not be sent...";
      }
?>
</body>
</html>
```

3.		**Attempt any THREE of the following:**	**12 M**
	a)	**Explain Indexed & Associative arrays with suitable example.**	**4 M**
	Ans	1. Indexed array - • An array with a numeric index where values are stored linearly. • Numeric arrays use number as access keys. • An access key is a reference to a memory slot in an array variable. • The access key is used whenever we want to read or assign a new value an array element. • **Syntax -** <?php $variable_name[n] = value; -?> OR <?php -$variable_name = array(n => value, …); -?> • **Example -** <?php $course = array(0 => "Computer Engg.",1 => "Information Tech.", o 2 => "Electronics and Telecomm."); echo $course[1]; ?> 2. Associative array - • This type of arrays is like the indexed arrays but instead of linear storage, every value can be assigned with a user-defined key of string type. • An array with a string index where instead of linear storage, each value can be assigned a specific key. • Associative array differ from numeric array in the sense that associative arrays	Each type explanation- 1 M Example of each type of array-1 M

		use descriptive names for id keys. • Syntax : <?php $variable_name['key_name'] = value; $variable_name = array('keyname' => value); ?>	
	b)	**Explain Constructor with example.**	**4 M**

	Ans	A constructor is a special built-in method.
		Constructor is special method of class used to initialize an object automatically when it is created.
		Constructor do not return any value.
		Constructor method takes arguments.
		To define constructor, 'construct' method is used with two underscores (__).
		Syntax:
		function __construct([argument1, argument2, ..., argumentN]) {
		/* Class initialization code */
		}
		Example:
		<?php
		class student{
		public $name;
		function __construct()
		{
		echo "Constructor is executed for " .$this->name;
		}
		function __destruct()
		{
		echo "Destructor is executing for " .$this->name;
		}
		}
		$s1=new student("Sneha");
		?>
			Explanation – 2 M
			Example – 2 M

	c)	**Define session & cookie. Explain use of session start.**	**4 M**
	Ans	Cookie: Cookie is a small piece of information which is stored at client browser. It is used to recognize the user.	Definition each – 1 M

- 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.

Use – 2 M

Session: Session data is stored on the server side and each Session is assigned with a unique Session ID (SID) for that session data. As session data is stored on the server there is no need to send any data along with the URL for each request to server.

Session_start-

- PHP session_start() function is used to start the session.
- It starts a new or resumes existing session.
- It returns existing session if session is created already.
- If session is not available, it creates and returns new session

Session variables are set with the PHP global variable: $_SESSION.

Example:

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

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

</body>
</html>
```

	d)	**Explain inserting and retrieving the query result operations.**	**4 M**
	Ans	The below given code will demonstrate to insert and retrieve the respective query result operations.	Relevant code – 4 M

```
<?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 = "SELECT id, firstname, lastname FROM staff";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
  // output data of each row
  while($row = $result->fetch_assoc()) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"].
"<br>";
  }
} else {
  echo "0 results";
}
$conn->close();
}
        ?>

<?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();
}
?>
```

4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Write a program to create PDF document in PHP.**	**4 M**

	Ans	<?php
		
		ob_end_clean();
		require('fpdf/fpdf.php');
		
		// Instantiate and use the FPDF class
			Relevant code – 4 M

```
$pdf = new FPDF();

//Add a new page
$pdf->AddPage();

// Set the font for the text
$pdf->SetFont('Arial', 'B', 18);

// Prints a cell with given text
$pdf->Cell(60,20,'Hello Students!');

// return the generated output
$pdf->Output();

?>
```

b) Write a program to demonstrate concept of inheritance in PHP. **4 M**

Ans

Relevant code – 4 M

Inheritance:

It is the process of inheriting (sharing) properties and methods of base class in its child class. Inheritance provides reusability of code in a program. PHP uses extends keyword to establish relationship between two classes.

Syntax:

```
class derived_class_name extends base_class_name

{

   // body of derived Class

 }
```

derived_class_name is the name of new class which is also known as child class and base_class_name is the name of existing class which is also known as parent class.

A derived class can access properties of base class and also can have its own properties. Properties defined as public in base class can be accessed inside as well as outside of the class but properties defined as protected in base class can be accessed only inside its derived class. Private members of class cannot be inherited.

Types of Inheritance are:

Single Inheritance

Multilevel Inheritance

Hierarchical Inheritance

Multiple Inheritance

Example:

```
<?php
class college
{
public $name="ABC College";
protected $code=7;
}
class student extends college
{
public $sname;
public function setName($n){
   $this->sname=$n;
}
public function display()
{
echo "College name=" .$this->name;
echo "<br>College code=" .$this->code;
echo "<br>Student name=" .$this->sname;
}
}
$s1=new student();
$s1->setName(“Ramesh”);
$s1->display();
```

		?> **OUTPUT:** College name=ABC college College code= 7 Student name=Ramesh	
	c)	**Create Employee form like employee_name, Address, Mobile_No, Date of birth, Post and Salary using different form input element and display user inserted values in new PHP form.**	**4 M**
	Ans	Html Program <!DOCTYPE html> <html lang="en"> <head> <title>Employee Form</title> </head> <body> Employee Information <form action="process_form.php" method="post"> <label for="employee_name">Employee Name:</label> <input type="text" id="employee_name" name="employee_name"> <label for="address">Address:</label> <input type="text" id="address" name="address"> <label for="mobile_no">Mobile No:</label> <input type="text" id="mobile_no" name="mobile_no"> <label for="dob">Date of Birth:</label> <input type="date" id="dob" name="dob"> <label for="post">Post:</label> <input type="text" id="post" name="post"> <label for="salary">Salary:</label> <input type="text" id="salary" name="salary"> <input type="submit" value="Submit"> </form> </body> </html>	Html code – 2 M PHP code – 2 M

PHP Program

```
<!DOCTYPE html>
<html lang="en">
<head>

   <title>Employee Information</title>
</head>
<body>
   <h2>Employee Information</h2>
   <?php
   // Retrieve form data using PHP
   $employee_name = $_POST['employee_name'];
   $address = $_POST['address'];
   $mobile_no = $_POST['mobile_no'];
   $dob = $_POST['dob'];
   $post = $_POST['post'];
   $salary = $_POST['salary'];

   // Display the retrieved data
   echo "<p><strong>Employee Name:</strong> $employee_name</p>";
   echo "<p><strong>Address:</strong> $address</p>";
   echo "<p><strong>Mobile No:</strong> $mobile_no</p>";
   echo "<p><strong>Date of Birth:</strong> $dob</p>";
   echo "<p><strong>Post:</strong> $post</p>";
   echo "<p><strong>Salary:</strong> $salary</p>";
   ?>
</body>
</html>
```

d) Write PHP program to demonstrate delete operation on table data. **4 M**

Ans In the below example a student record with a roll no. 'CO103' will be deleted by using DELETE statement and WHERE clause. Relevant code – 4 M

```
<?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 = "DELETE FROM staff WHERE id=1";

if ($conn->query($sql) === TRUE) {
  echo "Record deleted successfully";
```

		} else {
		echo "Error deleting record: " . $conn->error;
		}
		
		$conn->close();
		
		?>	

	e)	**Explain web page validation with example.**	**4 M**
	Ans	There are some built-in functions in PHP which will validate the web page. • User may by mistakenly submit the data through form with empty fields or in wrong format. • PHP script must ensure that required fields are complete and submitted data is in valid format. • PHP provides some inbuilt function using these functions that input data can be validated. • **empty()** function will ensure that text field is not blank it is with some data, function accepts a variable as an argument and returns **TRUE** when the text field is submitted with empty string, zero, NULL or FALSE value. • **Is_numeric()** function will ensure that data entered in a text field is a numeric value, the function accepts a variable as an argument and returns **TRUE** when the text field is submitted with numeric value. • **preg_match()** function is specifically used to performed validation for entering text in the text field, function accepts a "regular expression" argument and a variable as an argument which has to be in a specific pattern. Typically it is for validating email, IP address and pin code in a form. • **For Example** a PHP page formvalidation.php is having three text fields name, mobile number and email from user, on clicking Submit button a data will be submitted to PHP script validdata.php on the server, which will perform three different validation on these three text fields, it will check that name should not be blank, mobile number should be in numeric form and the email is validated with an email pattern. <html> <head> <title> Validating Form Data</title> </head> <body> <form method="post" action="validdata.php"> Name :<input type="text" name="name" id="name" /> Mobile Number :<input type="text" name="mobileno" id="mobileno" /> Email ID :<input type="text" name="email" id="email" /> <input type="submit" name="submit_btn" value="Submit" /> </form> </body> </html>	Explanation – 2 M Example – 2 M

```
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
if(empty($_POST['name']))
{
echo "Name can't be blank<br/>";
}
if(!is_numeric($_POST['mobileno']))
{
echo "Enter valid Mobile Number<br/>";
}
$pattern ='/\b[\w.-]+@[\w.-]+\.[A-Za-z]{2,6}\b/';
if(!preg_match($pattern,$_POST['email']))
{
echo "Enter valid Email ID.<br/>";
}
}
?>
```

5.		**Attempt any TWO of the following:**	**12 M**
	a)	**Explain decision making statements along with their syntax in PHP.**	**6 M**

	Ans	Program Instructions are executed sequentially. In some cases it is necessary to change the sequence of executions based on certain conditions. For this purpose decision control structure is required.
		
		Decision making statements are
		
		1. if statement
		2. if else statement
		3. Else if else statement
		4. Switch statement
		5. Break statement
		6. Continue statement
		
		If statement-
		
		The if statement executes some code if one condition is true.
		
		Syntax
		
		if (condition) {
		// code to be executed if condition is true;
			Each Syntax Carries – 1 M

}

Example-

```
<?php
$a=10;
if ($a > 0)
{
echo "The number is positive";
}
?>
```

1. If else statement –

The If...Else Statement

If you want to execute some code if a condition is true and another code if a condition is false, use the if ...else statement.

Syntax

```
if (condition)
  code to be executed if condition is true;
else
  code to be executed if condition is false;
```

```
<?php
    $d = date("D");
    if ($d == "Fri")
      echo "Have a nice weekend!";

    else
      echo "Have a nice day!";
  ?>
```

2. Nested if else statement-

   Nested if statements mean an if block inside another if block. Nested if else statement used when we have more than two conditions. It is also called if else if statement

```
if(condition1)
 {
// Code to be executed
 if condition1 is true
 }

 elseif(
condition2)
{
 // Code to be executed if the condition1 is false and condition2 is true
}
 else
{
// Code to be executed if both condition1 and condition2 are false
 }
```

4. The if...elseif...else statement executes different codes for more than two conditions.

Syntax

```
if (condition) {
  code to be executed if this condition is true;
} elseif (condition) {
  // code to be executed if first condition is false and this condition is true;
} else {
  // code to be executed if all conditions are false;
}
```

Example

Output "Have a good morning!" if the current time is less than 10, and "Have a good day!" if the current time is less than 20. Otherwise it will output "Have a

good night!":

```
<?php
$t = date("H");


if ($t < "10") {
  echo "Have a good morning!";
} elseif ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>
```

4.swich case –

The PHP switch Statement

Use the switch statement to select one of many blocks of code to be executed.

Syntax

```
switch (expression) {
  case label1:
    //code block
    break;
  case label2:
    //code block;
    break;
  case label3:
    //code block
    break;
  default:
```

```
    //code block
}
```

This is how it works:

The expression is evaluated once

The value of the expression is compared with the values of each case

If there is a match, the associated block of code is executed

The break keyword breaks out of the switch block

The default code block is executed if there is no match

```
<?php
$favcolor = "red";

switch ($favcolor) {
  case "red":
    echo "Your favorite color is red!";
    break;
  case "blue":
    echo "Your favorite color is blue!";
    break;
  case "green":
    echo "Your favorite color is green!";
    break;
  default:
    echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

4 Break : The keyword break ends execution of the current for, for each, while, do while or switch structure. When the

keyword break executed inside a loop the control automatically passes to the first

statement outside the loop. A

break is usually associated with the if.

Example :

```
<?php
for($i=1; $i<=5;$i++)
{
echo "$i<br/>";
if($i==3)
{
break;
}
}
?>
```

Output :

1

2

3

2. Continue : It is used to stop processing the current block of code in the loop and goes to the next iteration. It is used to skip a part of the body of the loop under certain conditions. It causes the loop to be continued with the next iteration after skipping any statement in between. The continue statement tells the compiler ”SKIP THE FOLLOWING STATEMENTS AND CONTINUE WITH THE NEXT ITERATION”.

Example :

```
<?php
for($i=1; $i<=5;$i++)
{
```

```
if($i==3)
{
continue;
}
echo "$i<br/>";
}
?>
```

Output :

1

2

4

5

	b)	**Write PHP program to demonstrate database operation using PHP and MySQL.**	**6 M**
	Ans	**Note - Assessor may consider any one database operation from this.** CREATE – A database consists of one or more tables. You will need special CREATE privileges to create or to delete a MySQL database. – A database is a collection of data. MySQL allows us to store and retrieve the data from the database in a efficient way.	Any One Correct Database Operation Carries 6 M (Create, Select, Insert, Update, Delete)

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

INSERT –

```
<?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 = "SELECT id, firstname, lastname FROM staff";
$result = $conn->query($sql);


if ($result->num_rows > 0) {
  // output data of each row
  while($row = $result->fetch_assoc()) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
  }
} else {
  echo "0 results";
}
$conn->close();
}
?>
```

DELETE-

- Record scan be deleted from a table using the SQL DELETE statement.
- DELETE statement is typically used in conjugation with the WHERE clause to delete only those records that matches specific criteria or condition.
- SQL query is formed using the DELETE statement and WHERE clause, after that will be executed by passing this query to the PHP query() function to delete the tables records.
- For example a student record with a roll no. ‘CO103’ will be deleted by using DELETE statement and WHERE clause.
- <?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);
}
$sql = "DELETE FROM staff WHERE id=1";
if ($conn->query($sql) === TRUE) {
 echo "Record deleted successfully";
} else {
 echo "Error deleting record: " . $conn->error;
}
$conn->close();
?>
```

INSERT-

After a database and a table have been created, we can start adding data in them.

Here are some syntax rules to follow:

The SQL query must be quoted in PHP

String values inside the SQL query must be quoted

Numeric values must not be quoted

The word NULL must not be quoted

The INSERT INTO statement is used to add new records to a MySQL table:

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

```
<?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();
}
?>
```

UPDATE-

Update Data In a MySQL Table Using MySQLi and PDO

The UPDATE statement is used to update existing records in a table:

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

```
<?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);
}
 $sql = "UPDATE staff SET lastname='technology' WHERE id=1";
if ($conn->query($sql) === TRUE) {
 echo "Record updated successfully";
}
else
{
 echo "Error updating record: " . $conn->error;
}
$conn->close();
}
 ?>
```

	c)	**Explain method overriding with example.**	**6 M**
	Ans	In function overriding, both parent and child classes should have same function name and number of arguments. It is used to replace the parent method in child class. The purpose of overriding is to change the behavior of parent class method. The two methods with the same name and same parameter is called overriding. Inherited methods can be overridden by redefining the methods (use the same name) in the child class. Example- <?php	Explanation- 2 M, Script- 4 M

```
class Shape
{
 public $length;
 public $width;
 public function __construct($length, $width) {
 $this->length = $length;
 $this->width = $width;
 }
 public function intro()
 {
 echo "The length is {$this->length} and the width is {$this->width}.";
 }
}
class square extends Shape
{
 public $height;
 public function __construct($length, $width, $height)
 {
 $this->length = $length;
 $this->width = $width;
$this->height = $height;
 }
 public function intro()
 {
 echo "The length is {$this->length}, the width is {$this->width}, the height is {$this->height} ";
```

		} } $s = new square(10,30, 50); $s->intro(); ?> Output : The length is 10, the width is 30, the height is 50	
6.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Write a PHP program to demonstrate use of cookies.**	**6 M**

	Ans	<html>
		<body>
		<?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!
		";
		echo "Value is: " . $_COOKIE[$cookie_name];
		}
		if(!isset($_COOKIE["user"]))
		{
		echo "Sorry, cookie is not found!";
		} else {
		echo "
		Cookie Value: " . $_COOKIE["user"];
		}
		setcookie("user"," ",time()-3600);
		echo "Cookie 'user' is deleted.";
		?>
			Correct logic- 6 M

		</body> </html>	
	b)	**Explain the following string functions with example:** **(i) Str-replace** **(ii) Vcwords()** **(ii) Strlen()** **(iv) Strtoupper()**	**6 M**

	Ans	**Note: in sub question (ii) instead of Vcwords() read it as ucwords()**
		1. str_replace()-Replaces some characters in a string (case-sensitive)
		Syntax- Str_replace(string tobe replaced, text, string)
		Example-
		<?php
		echo str_replace("Clock","Click","Click and Clock");
		?>
		Output:
		Click and Click
		
		2. ucwords()-Convert the first character of each word to uppercase.
		Syntax- ucwords(String)
		Example- <?php
		echo ucwords("welcome to php world");
		?>
		Output:
		Welcome To Php World
		
		3.strlen()-Returns the length of a string
		Syntax- Strlen(String)
		Example- <?php
		echo strlen("Welcome to PHP");
		?>
		Output: 14
		4. strtoupper()-Converts a string to uppercase letters
		Syntax - strtoupper(String)
		Example-
		<?php
		echo strtoupper("information technology ");
			Each Correct String Function Carries- 1.5 M

		Output: INFORMATION TECHNOLOGY	

	c)	**Explain interface with example.**	**6 M**

	Ans	An Interface allows the users to create programs, specifying the public methods that a class must implement, without involving the complexities and details of how the particular methods are implemented.
		– An Interface is defined just like a class but with the class keyword replaced by the interface keyword and just the function prototypes.
		– The interface contains no data variables.
		– An interface consists of methods that have no implementations, which means the interface methods are abstract methods.
		– All the methods in interfaces must have public visibility scope.
		-Interfaces are different from classes as the class can inherit from one class only whereas the class can implement one or more interfaces.
		– Interface enables you to model multiple inheritance
		
		Syntax-1 :
		
		interface (using class along with interface)
		
		class child_class_name extends parent_class_name
		
		implements interface_name1, ..
		
		Example :
		
		Example :
		
		<?php
		
		// Class A i.e. Parent A
		
		class A
		
		{
		
		public function disp1()
		
		{
		
		echo "Parent-A
		";
		}
			Interface explanation - 3 M, example-3 M

```
}
// Interface B i.e Parent B
interface B
{
public function disp2();
}
class C extends A implements B
{
function disp2()
 {
 echo " Parent-B <br>";
 }
 public function disp3()
 {
 echo "\nChild-C";
 }
}
$obj = new C();
$obj->disp1();
$obj->disp2();
$obj->disp3();
?>
```

Output :

```
Parent-A
Parent-B

 Child-C
```

Syntax-2 :

Interface (using multiple interface)

class child_class_name implements interface_name1,

interface_name2, ...

Example :

```
<?php
// interface A i.e. Parent A
interface A
{
 public function disp1();
}
// Interface B i.e Parent B
interface B
{
public function disp2();
}
class C implements A,B
{
function disp1()
 {
 echo "Parent-A <br>";
 }
function disp2()
 {
 echo " Parent-B <br>";
 }
```

```
 public function disp3()
 {
 }
}
$obj = new C();
$obj->disp1();
$obj->disp2();
$obj->disp3();
?>
```

Output :

Parent-A

Parent-B

Child-C