Question 1

1) Write a PHP program to demonstrate **Ternary**, **Null coalescing** and **Bitwise** Operators

PHP Program:

```
<!DOCTYPE html>
<html>
<head> <title>PHP Operators Demo</title>
</head>
<body>
<h2>PHP Operators Demo</h2>

<?php
// 1. Ternary Operator
$age = 20;
$result = ($age >= 18) ? "Adult" : "Minor";
echo "<h3>Ternary Operator:</h3>";
echo "Age is $age - Result: $result<br><br>";

// 2. Null Coalescing Operator
$name = null;
$username = $name ?? "Guest";
echo "<h3>Null Coalescing Operator:</h3>";
echo "Username is: $username<br><br>";

// 3. Bitwise Operators
$a = 5;  // 0101 in binary
$b = 3;  // 0011 in binary

echo "<h3>Bitwise Operators:</h3>";
echo "a = $a, b = $b<br>";
echo "a & b = " . ($a & $b) . "<br>"; // AND
echo "a | b = " . ($a | $b) . "<br>"; // OR
echo "a ^ b = " . ($a ^ $b) . "<br>"; // XOR
echo "~a = " . (~$a) . "<br>";        // NOT
echo "a << 1 = " . ($a << 1) . "<br>"; // Left Shift
echo "a >> 1 = " . ($a >> 1) . "<br>"; // Right Shift
?>

</body>
</html>
```

Explanation:

1. Ternary Operator (?:)

- Used as a shorthand for if-else.
- Syntax: condition ? value_if_true : value_if_false
- Here, we check if $age >= 18. If true, it returns "Adult", otherwise "Minor".

2. Null Coalescing Operator (??)

- Returns the left operand if it is **not null**, otherwise returns the right one.
- Here, $name is null, so $username becomes "Guest".

3. Bitwise Operators

- & (AND): Sets bit to 1 if both bits are 1.
- | (OR): Sets bit to 1 if any of the bits is 1.
- ^ (XOR): Sets bit to 1 if bits are different.
- ~ (NOT): Inverts all bits.
- << (Left Shift): Shifts bits to the left (multiplies by 2).
- >> (Right Shift): Shifts bits to the right (divides by 2).

---

2) Create a HTML form named **"multiformdemo.html"** that accepts two numbers from the user.(Include two labels: 'Enter first number' and 'Enter second number', along with two textboxes for number inputs, and four submit buttons)and Write PHP code to perform basic arithmetic operations **(addition, subtraction, multiplication, and division)** based on the button the user clicks.

Step 1: Create the HTML Form ("multiformdemo.html")

```
<!DOCTYPE html>
<html lang="en">
<head><title>Multi-Form Arithmetic Demo</title>
</head>
<body>
  <h2>Arithmetic Operations</h2>
  <form action="multiformdemo.php" method="post">
    <label for="num1">Enter first number:</label>
    <input type="number" name="num1" id="num1" required><br><br>

    <label for="num2">Enter second number:</label>
    <input type="number" name="num2" id="num2" required><br><br>

    <input type="submit" name="add" value="Add">
    <input type="submit" name="subtract" value="Subtract">
    <input type="submit" name="multiply" value="Multiply">
    <input type="submit" name="divide" value="Divide"><br><br>
  </form>
</body>
</html>
```

Step 2: PHP Code to Handle Arithmetic Operations ("multiformdemo.php")

```
<?php
// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get the input numbers from the form
    $num1 = $_POST['num1'];
    $num2 = $_POST['num2'];

    // Initialize a variable to store the result
    $result = "";

    // Perform operations based on the button clicked
    if (isset($_POST['add'])) {
        $result = $num1 + $num2;
    } elseif (isset($_POST['subtract'])) {
        $result = $num1 - $num2;
    } elseif (isset($_POST['multiply'])) {
        $result = $num1 * $num2;
    } elseif (isset($_POST['divide'])) {
        // Check if division by zero occurs
        if ($num2 != 0) {
            $result = $num1 / $num2;
        } else {
            $result = "Cannot divide by zero!";
        }
    }

    // Display the result
    echo "<h3>Result: $result</h3>";
}
?>
```

Explanation of the Code:

HTML Form (multiformdemo.html):

- The form has **two textboxes** to accept the **first number** and **second number** from the user.
- There are **four submit buttons** for different operations: **Add**, **Subtract**, **Multiply**, and **Divide**.
- The action attribute points to multiformdemo.php, which will handle the form submission and perform calculations.

PHP Code (multiformdemo.php):

- First, the code checks if the form has been **submitted** using $_SERVER["REQUEST_METHOD"] == "POST".
- It then fetches the numbers entered by the user using $_POST['num1'] and $_POST['num2'].

- Based on which button was clicked (isset($_POST['add']), etc.), it performs the corresponding arithmetic operation.
    - If the **divide** button is clicked, the code checks if the second number is **zero** (to prevent division by zero).
- Finally, the result of the calculation is displayed.

Question 2

1) Write a PHP program to demonstrate **Break** and **Continue** Statement

Explanation:

- **break**: It is used to exit or terminate the current loop, switch, or foreach statement prematurely.
- **continue**: It skips the remaining part of the current iteration in a loop and moves to the next iteration.

PHP Program to Demonstrate break and continue Statements

```
<?php
// Example of 'break' statement
echo "Using 'break' statement:<br>";
// Loop through numbers 1 to 10
for ($i = 1; $i <= 10; $i++) {
   // Break the loop when the number is 6
   if ($i == 6) {
      break;  // Exit the loop
   }
   echo "Number: $i<br>";
}

echo "<br>Using 'continue' statement:<br>";

// Example of 'continue' statement
// Loop through numbers 1 to 10
for ($i = 1; $i <= 10; $i++) {
   // Skip number 5
   if ($i == 5) {
      continue;  // Skip the current iteration and continue to the next
   }
   echo "Number: $i<br>";
}
?>
```

Explanation of the Code:

1. **break Statement**:
   - In the first loop (for ($i = 1; $i <= 10; $i++)), the program prints numbers from **1 to 5** and **breaks** the loop when the number reaches **6**. So, the loop exits at **6**, and no further numbers are printed.
2. **continue Statement**:
   - In the second loop (for ($i = 1; $i <= 10; $i++)), the program prints numbers from **1 to 10** except **5**. When the number is **5**, it **skips** the current iteration and continues to the next iteration, effectively excluding **5** from the output.

2) Create a HTML form named **“evenodddemo.html”** that accepts a number from the user.(Include a label 'Enter number', a textbox for number input, and a Submit button) and Write a PHP code to determine whether the given number is **odd** or **even**.

Step 1: HTML Form (evenodddemo.html)

```
<!DOCTYPE html>
<html>
<head>
   <title>Even Odd Check</title>
</head>
<body>
   <h2>Even or Odd Number Checker</h2>
   <form action="evenodddemo.php" method="post">
      <label>Enter number:</label>
      <input type="number" name="number" required>
      <br><br>
      <input type="submit" value="Check">
   </form>
</body>
</html>
```

- **action="evenodddemo.php" means the data will be sent to a PHP file for processing.**

Step 2: PHP File (evenodddemo.php)

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   $num = $_POST['number']; // Get number from form

   if ($num % 2 == 0) {
      echo "<h3>$num is Even</h3>";
   } else {
      echo "<h3>$num is Odd</h3>";
   }
}
?>
```

Explanation:

- $_POST['number']: Retrieves the number input from the form.
- $num % 2 == 0: Checks if the number is divisible by 2 (even).
- If not divisible, it is odd.

Question 3

1) Write a PHP program to display **even** numbers from 1-50 (using for ,while and do..while loop)

1. Program Code

```
<?php
//Using for loop
echo "<h3>Even Numbers using for loop:</h3>";
for ($i = 1; $i <= 50; $i++) {
    if ($i % 2 == 0) {
        echo $i . " ";
    }
}

//Using while loop
echo "<h3>Even Numbers using while loop:</h3>";
$i = 1;
while ($i <= 50) {
    if ($i % 2 == 0) {
        echo $i . " ";
    }
    $i++;
}

//Using do-while loop
echo "<h3>Even Numbers using do...while loop:</h3>";
$i = 1;
do {
    if ($i % 2 == 0) {
        echo $i . " ";
    }
    $i++;
} while ($i <= 50);
?>
```

Explanation:

- **Condition used:** $i % 2 == 0 checks if a number is even.
- All loops increment $i from 1 to 50.
- The even numbers are printed with a space using echo.

---

2) Write a PHP program to validate the name, email, password and mobile_no fields of html form

✅ **formvalidation.php – Single File Version**

```
<!DOCTYPE html>
<html>
<head>
   <title>Form Validation</title>
</head>
<body>

<h2>User Form</h2>
<form method="post">
   <label>Name:</label><br>
   <input type="text" name="name" required><br><br>

   <label>Email:</label><br>
   <input type="email" name="email" required><br><br>

   <label>Password:</label><br>
   <input type="password" name="password" required><br><br>

   <label>Mobile No:</label><br>
   <input type="text" name="mobile_no" required><br><br>

   <input type="submit" value="Submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   $name = trim($_POST["name"]);
   $email = $_POST["email"];
   $password = $_POST["password"];
   $mobile_no = $_POST["mobile_no"];

   $errors = [];

   // Validate Name
   if (empty($name) || !preg_match("/^[a-zA-Z\s]+$/", $name)) {
      $errors[] = "Invalid name. Only letters and spaces allowed.";
   }

   // Validate Email
   if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
      $errors[] = "Invalid email format.";
   }
   // Validate Password
   if (strlen($password) < 6) {
      $errors[] = "Password must be at least 6 characters.";
   }
```

```
    // Validate Mobile Number
    if (!preg_match("/^[0-9]{10}$/", $mobile_no)) {
        $errors[] = "Mobile number must be exactly 10 digits.";
    }

    // Show Result
    if (empty($errors)) {
        echo "<h3>Form Submitted Successfully!</h3>";
        echo "Name: $name<br>";
        echo "Email: $email<br>";
        echo "Mobile No: $mobile_no<br>";
    } else {
        echo "<h3>Form has errors:</h3>";
        foreach ($errors as $error) {
            echo "$error<br>";
        }
    }
}
?>

</body>
</html>
```

Question 4

1) Write a PHP program to display numbers from 1to 10 (using for ,while and do..while loop)

1. Program Code

```
<?php
//Using for loop
echo "<h3>Numbers from 1 to 10 using for loop:</h3>";
for ($i = 1; $i <= 10; $i++) {
   echo $i . " ";
}

//Using while loop
echo "<h3>Numbers from 1 to 10 using while loop:</h3>";
$i = 1;
while ($i <= 10) {
   echo $i . " ";
   $i++;
}

//Using do-while loop
echo "<h3>Numbers from 1 to 10 using do...while loop:</h3>";
$i = 1;
do {
   echo $i . " ";
   $i++;
} while ($i <= 10);
?>
```

---

2) Create customer form like customer name, address, mobile no, date of birth using different form of input elements and display user inserted values in new PHP form. – **Question seems wrong as how we can display values in php form? Makes no sense taking values and displaying it again in form, so correct question would be display in new php file! Still if this question comes in practical exam to you, first confirm from sir to display in new file or what?**

Step 1: HTML Form (customerform.html)

```
<!DOCTYPE html>
<html>
<head><title>Customer Form</title>
</head>
<body>
   <h2>Customer Details Form</h2>
   <form action="displaycustomer.php" method="post">
      <label>Customer Name:</label><br>
      <input type="text" name="name" required><br><br>
```

```
        <label>Address:</label><br>
        <textarea name="address" rows="4" cols="30" required></textarea><br><br>

        <label>Mobile No:</label><br>
        <input type="text" name="mobile" required><br><br>

        <label>Date of Birth:</label><br>
        <input type="date" name="dob" required><br><br>

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

Step 2: PHP File to Display Data (displaycustomer.php)

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['name'];
    $address = $_POST['address'];
    $mobile = $_POST['mobile'];
    $dob = $_POST['dob'];

    echo "<h2>Customer Information</h2>";
    echo "Name: " . $name . "<br>";
    echo "Address: " . $address . "<br>";
    echo "Mobile No: " . $mobile . "<br>";
    echo "Date of Birth: " . $dob . "<br>";
}
?>
```

Explanation:

- We use different input types:
    - text for name and mobile number
    - textarea for address
    - date for date of birth
- On form submission, values are **sent using POST method** to displaycustomer.php.
- The PHP file reads and prints the values using $_POST.

Question 5

1) Write a PHP program to display **factorial** of number (using for ,while and do..while loop).

1. Program Code :

```
<?php
$number = 5; //number for factorial finding

//Using for loop
$factorial = 1;
for ($i = 1; $i <= $number; $i++) {
   $factorial *= $i;
}

echo "<h3>Factorial of $number using for loop is: $factorial</h3>";

//Using while loop
$factorial = 1;
$i = 1;
while ($i <= $number) {
   $factorial *= $i;
   $i++;
}

echo "<h3>Factorial of $number using while loop is: $factorial</h3>";

//Using do-while loop
$factorial = 1;
$i = 1;
do {
   $factorial *= $i;
   $i++;
} while ($i <= $number);

echo "<h3>Factorial of $number using do...while loop is: $factorial</h3>";
?>
```

Explanation:

- **Factorial** means multiplying the number by all integers below it till 1.
    - Example: 5! = 5 × 4 × 3 × 2 × 1 = 120
- We use a loop to multiply the numbers from 1 to $number.
- *= is shorthand for multiplication and assignment.

2) Write a PHP programs to create database, create table and insert records in a table.

Program Code : Create DB, Create Table, Insert Record

```
<?php
$servername = "localhost";
$username = "root";
$password = "";

// Step 1: Create Database
$conn = new mysqli($servername, $username, $password);
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$sql = "CREATE DATABASE IF NOT EXISTS mydemoDB";
if ($conn->query($sql) === TRUE) {
   echo "Database created successfully.<br>";
} else {
   die("Error creating database: " . $conn->error);
}
$conn->close();

// Step 2: Create Table in the New Database
$conn = new mysqli($servername, $username, $password, "mydemoDB");
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$sql = "CREATE TABLE IF NOT EXISTS customers (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(50),
   email VARCHAR(50),
   mobile VARCHAR(15)
)";
if ($conn->query($sql) === TRUE) {
   echo "Table 'customers' created successfully.<br>";
} else {
   die("Error creating table: " . $conn->error);
}

// Step 3: Insert a Sample Record
$sql = "INSERT INTO customers (name, email, mobile)
       VALUES ('Jay', 'jay@example.com', '9876543210')";
if ($conn->query($sql) === TRUE) {
   echo "Record inserted successfully.";
} else {
   echo "Error inserting record: " . $conn->error;
}

$conn->close();
?>
```

Question 6

1) Write a PHP program to find largest of three numbers

PHP Code:

```
<?php
$num1 = 25;
$num2 = 40;
$num3 = 15;

if ($num1 >= $num2 && $num1 >= $num3) {
   echo "The largest number is: $num1";
} elseif ($num2 >= $num1 && $num2 >= $num3) {
   echo "The largest number is: $num2";
} else {
   echo "The largest number is: $num3";
}
?>
```

Explanation:

- We compare the numbers using >= (greater than or equal to).
- If num1 is greater than or equal to both num2 and num3, it's the largest.
- Else if num2 is greater than or equal to the others, it's the largest.
- Otherwise, num3 is the largest.
- This also handles the **case where two or more numbers are equal**.

---

2) Write a PHP program to Insert data into employee table and Select data from employee table.

Program Code :

```
<?php
// Connect to database, you can take variables as well for this values
$conn = mysqli_connect("localhost", "root", "", "mydemoDB");

if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}

// --- Step 1: Insert a new employee record ---
$name = "Jay";
$email = "jay@example.com";
$department = "IT";

$insert_sql = "INSERT INTO employee (name, email, department)
            VALUES ('$name', '$email', '$department')";
```

```
if (mysqli_query($conn, $insert_sql)) {
    echo "<p>✅ Record inserted successfully</p>";
} else {
    echo "<p>❌ Error inserting record: " . mysqli_error($conn) . "</p>";
}

// --- Step 2: Fetch and display all records ---
$select_sql = "SELECT * FROM employee";
$result = mysqli_query($conn, $select_sql);

if (mysqli_num_rows($result) > 0) {
    echo "<h3>📋 Employee Records:</h3>";
    echo "<table border='1' cellpadding='5'>
          <tr><th>ID</th><th>Name</th><th>Email</th><th>Department</th></tr>";
    while ($row = mysqli_fetch_assoc($result)) {
        echo "<tr>"
        echo "<td>" . $row['id'] . "</td>";
        echo "<td>" . $row['name'] . "</td>";
        echo "<td>" . $row['email'] . "</td>";
        echo "<td>" . $row['department'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
} else {
    echo "<p>No employee records found.</p>";
}

mysqli_close($conn);
?>
```

Question 7

1) Write a calendar program using switch statement.

PHP Code:

```
<?php
$dayNumber = 4; // You can change this to any number from 1 to 7

switch ($dayNumber) {
   case 1:
      echo "It's Monday";
      break;
   case 2:
      echo "It's Tuesday";
      break;
   case 3:
      echo "It's Wednesday";
      break;
   case 4:
      echo "It's Thursday";
      break;
   case 5:
      echo "It's Friday";
      break;
   case 6:
      echo "It's Saturday";
      break;
   case 7:
      echo "It's Sunday";
      break;
   default:
      echo "Invalid input. Enter a number between 1 and 7.";
}
?>
```

Explanation:

- $dayNumber stores a number from 1 to 7.
- switch checks which number it is and displays the correct **day name**.
- default case handles wrong inputs.

2) Write a PHP program to Update and Delete data from employee table

Program Code :

```
<?php
// Connect to database
$conn = mysqli_connect("localhost", "root", "", "mydemoDB");

if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}

// --- Step 1: Update employee with ID 1 ---
$update_sql = "UPDATE employee
            SET email = 'newemail@example.com', department = 'HR'
            WHERE id = 1";

if (mysqli_query($conn, $update_sql)) {
   echo "<p>Record with ID 1 updated successfully.</p>";
} else {
   echo "<p> Error updating record: " . mysqli_error($conn) . "</p>";
}

// --- Step 2: Delete employee with ID 2 ---
$delete_sql = "DELETE FROM employee WHERE id = 2";

if (mysqli_query($conn, $delete_sql)) {
   echo "<p>Record with ID 2 deleted successfully.</p>";
} else {
   echo "<p>Error deleting record: " . mysqli_error($conn) . "</p>";
}

mysqli_close($conn);
?>
```

Question 8

1) Write a PHP program creating and traversing an Indexed, Associative and Multidimensional array(using for loop or foreach loop)

PHP Code: Array Creation & Traversal

```
<?php
// 1. Indexed Array
$fruits = ["Apple", "Banana", "Mango", "Orange"];
echo "<h4>Indexed Array:</h4>";
for ($i = 0; $i < count($fruits); $i++) {
   echo $fruits[$i] . "<br>";
}

// 2. Associative Array
$student = ["name" => "Jay",  "age" => 21,  "course" => "BCA"];
echo "<h4>Associative Array:</h4>";
foreach ($student as $key => $value) {
   echo "$key : $value <br>";
}

// 3. Multidimensional Array
$employees = [
   ["id" => 1, "name" => "Amit", "department" => "IT"],
   ["id" => 2, "name" => "Sneha", "department" => "HR"],
   ["id" => 3, "name" => "Jay", "department" => "Finance"]
];
echo "<h4>Multidimensional Array:</h4>";
foreach ($employees as $employee) {
   foreach ($employee as $key => $value) {
      echo "$key : $value <br>";
   }
   echo "<hr>"; // Just a line break between employees
}
?>
```

Explanation:

1. Indexed Array

- Elements are accessed by numeric index (0, 1, 2...).
- We used a for loop with count() to iterate over all items.

2. Associative Array

- Uses named keys instead of index.
- We used foreach to get each key–value pair.

3. Multidimensional Array

- An array of associative arrays.
- We used **nested foreach** to loop through each employee and print their details.

2) Write a PHP program for sending mail

Note: Follow configuration steps given by sir before running the code.

PHP Code: sendmail.php

```
<?php
$to = "receiver@example.com";          // Receiver's email address use original
$subject = "Test Email from PHP";      // Email subject
$message = "This is a test mail sent using PHP's mail() function.";  // Email body
$headers = "From: sender@example.com";  // Sender's email use original

if (mail($to, $subject, $message, $headers)) {
   echo "Email sent successfully.";
} else {
   echo "Failed to send email.";
}
?>
```

Explanation:

- **$to** – Recipient's email address.
- **$subject** – The subject of the email.
- **$message** – The message body.
- **$headers** – Optional headers like sender email.

The mail() function sends the email and returns true if successful.

Question 9

1) Write a PHP program to demonstrate **Sort functions** for **Arrays**(sort(),rsort(),asort() , ksort() , arsort(),krsort())

PHP Program :

```
<?php
// Indexed Array
$numbers = [5, 2, 8, 1, 9];
echo "<h4>Original Indexed Array:</h4>";
print_r($numbers);

// sort() - Ascending order
sort($numbers);
echo "<h4>After sort():</h4>";
print_r($numbers);

// rsort() - Descending order
rsort($numbers);
echo "<h4>After rsort():</h4>";
print_r($numbers);

// Associative Array
$student = ["name" => "Jay",  "age" => 21, "marks" => 85];
echo "<h4>Original Associative Array:</h4>";
print_r($student);

// asort() - Sort by value (ascending), keep key association
asort($student);
echo "<h4>After asort():</h4>";
print_r($student);

// arsort() - Sort by value (descending), keep key association
arsort($student);
echo "<h4>After arsort():</h4>";
print_r($student);

// ksort() - Sort by key (ascending)
ksort($student);
echo "<h4>After ksort():</h4>";
print_r($student);

// krsort() - Sort by key (descending)
krsort($student);
echo "<h4>After krsort():</h4>";
print_r($student);
?>
```

Explanation of Functions:

Function	Description
sort()	Sorts indexed array in ascending order (values only, keys reset)
rsort()	Sorts indexed array in descending order (values only, keys reset)
asort()	Sorts associative array by values in ascending order (keeps keys)
arsort()	Sorts associative array by values in descending order (keeps keys)
ksort()	Sorts associative array by keys in ascending order
krsort()	Sorts associative array by keys in descending order

---

2) Write a PHP program to count total number of rows in the database table

PHP Code:

```
<?php
// Connect to database
$conn = mysqli_connect("localhost", "root", "", "mydemoDB");

// Check connection
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}

// SQL query to count total rows
$sql = "SELECT COUNT(*) AS total FROM employee";
$result = mysqli_query($conn, $sql);

// Fetch and display the count
if ($row = mysqli_fetch_assoc($result)) {
   echo "Total number of rows in 'employee' table: " . $row['total'];
} else {
   echo "Error fetching row count.";
}

// Close the connection
mysqli_close($conn);
?>
```

Explanation:

- COUNT(*) counts all rows in the table.
- We use **AS total** to give an alias name to the count column.
- mysqli_fetch_assoc() is used to access the result as an associative array.

Question 10

1) Write a PHP program to demonstrate use of various built-in string functions

Try to use at least 5-7 functions in exam

PHP Code:

```
<?php
$str = " Hello World! ";

// 1. strlen() – Length of string
echo "1. Length: " . strlen($str) . "<br>";

// 2. str_word_count() – Count number of words
echo "2. Word count: " . str_word_count($str) . "<br>";

// 3. strrev() – Reverse the string
echo "3. Reverse: " . strrev($str) . "<br>";

// 4. strtolower() – Convert to lowercase
echo "4. Lowercase: " . strtolower($str) . "<br>";

// 5. strtoupper() – Convert to uppercase
echo "5. Uppercase: " . strtoupper($str) . "<br>";

// 6. trim() – Remove spaces from both ends
echo "6. Trimmed: '" . trim($str) . "'<br>";

// 7. ltrim() and rtrim() – Remove spaces from left/right
echo "7. Left Trim: '" . ltrim($str) . "'<br>";
echo "8. Right Trim: '" . rtrim($str) . "'<br>";

// 8. strpos() – Find position of substring
echo "9. Position of 'World': " . strpos($str, "World") . "<br>";

// 9. str_replace() – Replace text
echo "10. Replace 'World' with 'Jay': " . str_replace("World", "Jay", $str) . "<br>";

// 10. substr() – Extract part of string
echo "11. Substring (6 to 11): " . substr($str, 6, 5) . "<br>";
?>
```

Explanation of Functions:

Function	Purpose
strlen()	Returns length of the string
str_word_count()	Counts total words

strrev()	Reverses the string
strtolower()	Converts to lowercase
strtoupper()	Converts to uppercase
trim()	Removes whitespace from both ends
ltrim() / rtrim()	Removes whitespace from left/right
strpos()	Finds the position of a substring
str_replace()	Replaces part of the string
substr()	Extracts part of a string

---

2) Write a PHP programs to implements Single, Multilevel and Multiple inheritances.

PHP Program :

```
<?php
echo "<h2>1. Single Inheritance</h2>";

class Animal {
   public function sound() {
      echo "Animals make sound<br>";
   }
}

class Dog extends Animal {
   public function bark() {
      echo "Dog barks<br>";
   }
}

$dog = new Dog();
$dog->sound(); // Inherited method
$dog->bark();  // Own method


echo "<h2>2. Multilevel Inheritance</h2>";

class Grandfather {
   public function house() {
      echo "Grandfather's house<br>";
   }
}

class Father extends Grandfather {
   public function car() {
      echo "Father's car<br>";
```

```
    }
}

class Son extends Father {
    public function bike() {
        echo "Son's bike<br>";
    }
}

$son = new Son();
$son->house();
$son->car();
$son->bike();


echo "<h2>3. Multiple Inheritance using Traits</h2>";

trait Painter {
    public function draw() {
        echo "Drawing a picture<br>";
    }
}

trait Singer {
    public function sing() {
        echo "Singing a song<br>";
    }
}

class Artist {
    use Painter, Singer;

    public function showTalent() {
        echo "I have multiple talents<br>";
    }
}

$artist = new Artist();
$artist->draw();
$artist->sing();
$artist->showTalent();
?>
```

🧠 **Summary:**

Type of Inheritance	PHP Support	How it's Done
Single	✅ Yes	Using extends
Multilevel	✅ Yes	Using multiple levels of extends
Multiple	⚠️ Not Directly	Use traits

Question 11

1) Write a PHP programs to calculate length of string and count number of words in string without using string functions.

Counting words is bit tricky, in solution what we are doing is similar to calculate length we are traversing through entire loop by checking each indexed string value whether set or not (isset($string[$i])) , then in if block we are checking whether current character is not a whitespace and we are not already in word which means suppose sentence is "hey there" and we are somewhere 't' so currently its not whitespace but its in word means that we are in 'there' word so its not new word in a sentence so no word count increment needed only increment when the current character is not white space and we are not in word

✅ **PHP Code: stringlength_wordcount.php**

```
<?php
$string = "Hello this is a demo string";

// Calculate length without using strlen()
$length = 0;
for ($i = 0; isset($string[$i]); $i++) {
    $length++;
}
echo "Length of string: $length <br>";

// Count words without using str_word_count()
$wordCount = 0;
$inWord = false;

for ($i = 0; isset($string[$i]); $i++) {
    if ($string[$i] != ' ' && !$inWord) {
        $wordCount++;
        $inWord = true;
    } elseif ($string[$i] == ' ') {
        $inWord = false;
    }
}
echo "Number of words: $wordCount";
?>
```

Explanation:

- **Length:**
    - Loop through each character.

- Use isset($string[$i]) to safely check character existence.
- Increase counter manually — no strlen() used.

- **Word Count:**
  - Count transitions from space → non-space.
  - When you hit a non-space after a space (or at the start), it's a new word.
  - Simple, clean logic — no str_word_count().

---

2) Develop a PHP programs to demonstrate function overloading and overriding.

PHP Program :

```
<?php
echo "<h2>1. Function Overloading (using __call)</h2>";

class Calculator {
   public function __call($name, $arguments) {
        if ($name == "add") {
                $sum = 0;
                foreach ($arguments as $num) {
                   $sum += $num;
                }
                echo "Sum of " . count($arguments) . " numbers: $sum<br>";
        }
     }
   }
}

$calc = new Calculator();
$calc->add(10, 20);        // Output: Sum of 2 numbers: 30
$calc->add(5, 10, 15);     // Output: Sum of 3 numbers: 30


echo "<h2>2. Function Overriding</h2>";
class Animal {
   public function makeSound() {
      echo "Animal makes sound<br>";
   }
}

class Dog extends Animal {
   public function makeSound() {
      echo "Dog barks<br>";
   }
}
```

```
$animal = new Animal();
$animal->makeSound(); // Output: Animal makes sound

$dog = new Dog();
$dog->makeSound();    // Output: Dog barks (overridden)
?>
```

Output Preview:

1. Function Overloading

Sum of 2 numbers: 30

Sum of 3 numbers: 30

2. Function Overriding

Animal makes sound

Dog barks

Question 12

1) Write a PHP programs to demonstrate Parameterized function, Variable function and Anonymous function

```
<?php
//1. Parameterized Function
//A function that accepts parameters to perform some operation.

function greet($name) {
   echo "Hello, $name!<br>";
}

greet("Jay");
greet("Amit");

//2. Variable Function
//You can assign a function name to a variable and then call it using that variable.

function welcome() {
   echo "Welcome to PHP world!<br>";
}

$myFunction = "welcome"; // storing function name in variable
$myFunction();          // calling using variable

//3. Anonymous Function (Function without a name)
//You assign the function to a variable and then call it.

$add = function($a, $b) {
   return $a + $b;
};

echo "Sum = " . $add(10, 20); // calling the anonymous function
?>
```

Summary Table

Concept	What it is	Example Call
Parameterized	Function with parameters	greet("Jay")
Variable Function	Function called via variable, function name stored in variable as string value and called using that variable	$myFunction();
Anonymous Function	Function with no name, stored in variable	$add(10, 20);

2) Write PHP program for cloning of an object

What is Object Cloning in PHP?

In PHP, **cloning** means creating a **copy of an object.**

We do it using the `**clone**` keyword.

Example Code:

```
<?php
class Student {
   public $name; public $roll;
   public function setDetails($name, $roll) {
      $this->name = $name;
      $this->roll = $roll;
   }
   public function showDetails() {
      echo "Name: $this->name <br> Roll No: $this->roll <br><br>";
   }
}

// Original object
$student1 = new Student();
$student1->setDetails("Jay", 101);
echo "Original Object:<br>";
$student1->showDetails();

// Cloning the object
$student2 = clone $student1;
$student2->setDetails("Amit", 102); // Modify cloned object
echo "Cloned Object (Modified):<br>";
$student2->showDetails();
?>
```

Output:

Original Object:
Name: Jay
Roll No: 101

Cloned Object (Modified):
Name: Amit
Roll No: 102

Explanation:

Code	**Meaning**
$student1 = new Student();	Creating the original object
$student2 = clone $student1;	Cloning `$student1` into a new object `$student2`
$student2->setDetails("Amit", 102);	Changing the values of the cloned object

✔ This proves they are separate objects — changes in one do **not** affect the other.

Question 13

1) Write a PHP programs to draw a rectangle filled with red color and to display text on image

PHP Program to Draw a Red Rectangle and Display Text on the Image
To create and manipulate images in PHP, we use the **GD library**. Make sure the GD library is enabled in your PHP installation.

Program Code:

```
<?php
// Create a blank image (width: 400px, height: 200px)
$image = imagecreatetruecolor(400, 200);

// Allocate colors
$red = imagecolorallocate($image, 255, 0, 0); // Red color for the rectangle
$white = imagecolorallocate($image, 255, 255, 255); // White color for the text

// Fill the rectangle with red color
imagefilledrectangle($image, 50, 50, 350, 150, $red);

// Add some text in the center of the rectangle
$text = "Hello, Welcome!";
$font = 5; // Built-in font size 5
imagestring($image, $font, 120, 90, $text, $white);

// Set the content type for the browser (image format)
header('Content-Type: image/png');

// Output the image to the browser
imagepng($image);

// Free up memory – not necessary step
imagedestroy($image);
?>
```

Explanation:

1. **imagecreatetruecolor(400, 200)**:

   Creates a blank image with dimensions **400px x 200px**.

2. **imagecolorallocate($image, 255, 0, 0)**:

   Allocates the color **red** to be used for drawing the rectangle.

3. **imagefilledrectangle($image, 50, 50, 350, 150, $red)**:

   Draws a **filled red rectangle** starting from the point (50, 50) to (350, 150).

4. **imagestring($image, $font, 120, 90, $text, $white)**:
   Places the text "Hello, Welcome!" inside the image, with the color **white** at the position (120, 90).
5. **header('Content-Type: image/png')**:
   Sends the appropriate HTTP header to the browser so it knows the content is an image.
6. **imagepng($image)**:
   Outputs the image in **PNG** format to the browser.
7. **imagedestroy($image)**:
   Frees the memory associated with the image after it's been output.

✅ **Output:**

When you run this PHP script, it will display an image with a red rectangle and the text "Hello, Welcome!" on it.

⚡ **Notes:**

- This program uses PHP's **GD library**, which is available by default in most PHP installations.
- **Font size 5** is used for simplicity here. You can experiment with other font sizes

---

2) Write a PHP programs to demonstrate Serialization and Introspection

Sure Jay! Let's break this down.

Serialization and Introspection in PHP:

1. **Serialization**:
   - Serialization is the process of converting a PHP object or data into a storable or transmittable format (usually a string).
   - This is useful when you want to store or send an object or array as a string, then later retrieve it back.
2. **Introspection**:
   - Introspection is the ability to examine the structure of a class or object at runtime. This allows you to inspect its properties, methods, and other details about the class.
   - PHP provides functions like get_class(), get_object_vars(), get_class_methods(), etc., to introspect an object.

1. Serialization

```
<?php
class Person {
    public $name;
    public $age;
    public function __construct($name, $age) {
        $this->name = $name;
        $this->age = $age;
    }
    // Method to display object details
    public function display() {
        echo "Name: " . $this->name . "<br>";
        echo "Age: " . $this->age . "<br>";
    }
}

// Create an object of Person
$person1 = new Person("Jay", 22);

// Serialize the object
$serializedPerson = serialize($person1);
echo "Serialized Object: " . $serializedPerson . "<br>";

// Unserialize the object
$unserializedPerson = unserialize($serializedPerson);

// Display the unserialized object
echo "Unserialized Object: <br>";
$unserializedPerson->display();
?>
```

Explanation of Serialization:

1. **serialize($person1)**:

   Converts the $person1 object into a string format.

2. **unserialize($serializedPerson)**:

   Restores the original object from the serialized string.

3. **display()**:

   Displays the properties of the object (name and age).

Output:

Serialized Object: O:6:"Person":2:{s:4:"name";s:3:"Jay";s:3:"age";i:22;}

Unserialized Object:

Name: Jay

Age: 22

2. Introspection

```
<?php
class Person {
    public $name;
    public $age;
    public function __construct($name, $age) {
        $this->name = $name;
        $this->age = $age;
    }
    // Method to display object details
    public function display() {
        echo "Name: " . $this->name . "<br>";
        echo "Age: " . $this->age . "<br>";
    }
}

// Create an object of Person
$person2 = new Person("Amit", 25);

// Introspection: Get the class name
echo "Class Name: " . get_class($person2) . "<br>";

// Introspection: Get the class properties (public variables)
echo "Class Properties: <br>";
print_r(get_object_vars($person2));

echo "<br>";

// Introspection: Get all methods of the class
echo "Class Methods: <br>";
print_r(get_class_methods($person2));
?>
```

Explanation of Introspection:

1. **get_class($person2)**:

   Returns the name of the class (Person in this case).

2. **get_object_vars($person2)**:

   Returns an associative array of all public properties of the object.

3. **get_class_methods($person2)**:

   Returns an array of all method names in the class.

Output:

```
Class Name: Person
Class Properties:
```

```
Array
(
    [name] => Amit
    [age] => 25
)
Class Methods:
Array
(
    [0] => display
)
```

Key Takeaways:

1. **Serialization**:
    - Converts an object or data into a string format and later restores it back.
    - Useful for storing objects in sessions or databases or sending them over networks.
2. **Introspection**:
    - Allows you to inspect the class and object structure at runtime.
    - You can easily retrieve class names, properties, and methods using built-in PHP functions.

Question 14

1) Write a PHP program to create a class as “Rectangle” with two properties length and width.

   Calculate area and perimeter of rectangle for two objects

   PHP Program: Rectangle Class

```
<?php
class Rectangle {
    public $length;
    public $width;
    // Constructor to initialize length and width
    public function __construct($length, $width) {
        $this->length = $length;
        $this->width = $width;
    }

    // Method to calculate area
    public function getArea() {
        return $this->length * $this->width;
    }

    // Method to calculate perimeter
    public function getPerimeter() {
        return 2 * ($this->length + $this->width);
    }
}

// Create first rectangle object
$rect1 = new Rectangle(10, 5);
echo "Rectangle 1:<br>";
echo "Area: " . $rect1->getArea() . "<br>";
echo "Perimeter: " . $rect1->getPerimeter() . "<br><br>";

// Create second rectangle object
$rect2 = new Rectangle(8, 6);
echo "Rectangle 2:<br>";
echo "Area: " . $rect2->getArea() . "<br>";
echo "Perimeter: " . $rect2->getPerimeter();
?>
```

Sample Output:

Rectangle 1:
Area: 50
Perimeter: 30

Rectangle 2:
Area: 48
Perimeter: 28

2) Write a PHP program to create access, modify and delete a cookie.

PHP Program

```
<?php
// 1. Create a cookie
setcookie("username", "Jay", time() + 3600); // valid for 1 hour
echo "Cookie 'username' created with value 'Jay'<br>";

// 2. Access the cookie
if (isset($_COOKIE['username'])) {
   echo "Accessed Cookie Value: " . $_COOKIE['username'] . "<br>";
} else {
   echo "Cookie not available yet (refresh the page to access it)<br>";
}

// 3. Modify the cookie
setcookie("username", "JayUpdated", time() + 3600); // modify it
echo "Cookie 'username' modified to 'JayUpdated'<br>";

// 4. Delete the cookie
setcookie("username", "", time() - 3600); // delete it
echo "Cookie 'username' deleted<br>";
?>
```

How it Works (Important Notes):

- When this script runs the **first time**, it **creates** the cookie and **modifies** and **deletes** it immediately.
- Since **cookie changes only take effect on the next request**, you'll need to **refresh** the page to properly **access** or confirm if the cookie was modified/deleted.
- This is normal behavior due to how HTTP cookies work — they are sent by the browser in the **next** request.

What You'll See:

On first run:

Cookie 'username' created with value 'Jay'

Cookie not available yet (refresh the page to access it)

Cookie 'username' modified to 'JayUpdated'

Cookie 'username' deleted

On refreshing:

Cookie 'username' created with value 'Jay'

Accessed Cookie Value: Jay

Cookie 'username' modified to 'JayUpdated'

Cookie 'username' deleted

Question 15

1) Write a PHP programs to demonstrate default constructor, parameterized constructor and destructor

PHP Program: Constructors & Destructor

```
<?php
// Class with Default Constructor
class DefaultConstructor {
    public function __construct() {
        echo "Default Constructor is called<br>";
    }
}

// Class with Parameterized Constructor
class ParameterizedConstructor {
    public function __construct($name, $age) {
        //set properties here such as $name and $age
        echo "Parameterized Constructor is called<br>";
        echo "Name: $name<br>";
        echo "Age: $age<br>";
    }
}

// Class with Destructor
class DestructorDemo {
    public function __construct() {
        echo "Constructor of DestructorDemo called<br>";
    }
    public function __destruct() {
        echo "Destructor is called when object is destroyed<br>";
    }
}

// Create object of DefaultConstructor
$default = new DefaultConstructor();
echo "<hr>";

// Create object of ParameterizedConstructor
$param = new ParameterizedConstructor("Jay", 21);
echo "<hr>";

// Create and use DestructorDemo
$demo = new DestructorDemo();
echo "Doing some work in DestructorDemo object...<br>";
unset($demo); // Force object destruction here
?>
```

💡 **Output:**

```
Default Constructor is called
---------------------------
Parameterized Constructor is called
Name: Jay
Age: 21
---------------------------
Constructor of DestructorDemo called
Doing some work in DestructorDemo object...
Destructor is called when object is destroyed
```

Summary for Viva:

- You used __construct() in different classes for default and parameterized.
- PHP does not allow multiple constructors in a single class (no overloading).
- Destructor is written using __destruct() and is called when object is destroyed or script ends.

---

2) Write a PHP program to start session, set session variables, get session variables, modify session Variables and destroy session

PHP Program for Session Handling :

```
<?php
// Start the session
session_start();

// 1. Set session variables
$_SESSION["username"] = "Jay";
$_SESSION["role"] = "Student";
echo "Session variables are set.<br>";

// 2. Access session variables
echo "Username: " . $_SESSION["username"] . "<br>";
echo "Role: " . $_SESSION["role"] . "<br>";

// 3. Modify session variables
$_SESSION["username"] = "JayModified";
echo "Username after modification: " . $_SESSION["username"] . "<br>";

// 4. Destroy the session (uncomment below line to test destroying)
// session_destroy();
// echo "Session destroyed.";
?>
```

Explanation:

- session_start()

  Must be called before any output to start or resume a session.
- $_SESSION["key"] = "value";

  Used to **set** or **modify** session variables.
- echo $_SESSION["key"];

  Used to **access** session variables.
- session_destroy();

  Used to **destroy** the session (removes all session variables).

  Note: After session_destroy(), refresh is needed to see it's gone.

✅ **Output:**

Session variables are set.

Username: Jay

Role: Student

Username after modification: JayModified

If you **uncomment** session_destroy(), then after refreshing, variables will be gone.

.