1. Write a PHP scripts that demonstrate fundamentals PHP Prime number

```
<!DOCTYPE html>
<html>
<head>
<title>Check Prime Number</title>
</head>
<body>
   <form action="" method="get">
      Enter a number:
      <input type="number" name="number">
      <input type="submit" value="Check">
   </form>
   <?php
   function isPrime($num) {
      // 1 is not a prime number
      if ($num <= 1) {
         return false;
      }
      // Check from 2 to square root of the number
      for ($i = 2; $i <= sqrt($num); $i++) {
         if ($num % $i == 0) {
            // If the number is divisible by any number between 2 and its square root, it is not prime
            return false;
         }
      }
      return true;
   }
   if (isset($_GET['number'])) {
      $number = $_GET['number'];
```

```
        if (is_numeric($number)) {
            if (isPrime($number)) {
                echo "$number is a prime number.";
            } else {
                echo "$number is not a prime number.";
            }
        } else {
            echo "Please enter a valid number.";
        }
    }
    ?>
</body>
</html>
```

Output:

If you enter 7: -

7 is a prime number.

If you enter 12: -

12 is not a prime number.

If you enter non-numeric input: -

Please enter a valid number.

2. Write PHP scripts that demonstrate fundamentals PHP Factorial.

```
<!DOCTYPE html>
<html>
<head>
<title>Factorial Calculator</title>
</head>
<body>
    <form action="" method="get">
        Enter a number:
        <input type="number" name="number">
        <input type="submit" value="Calculate">
    </form>
    <?php
    function factorial($num) {
        if ($num < 0) {
            return "Factorial is not defined for negative numbers.";
        } elseif ($num == 0) {
            return 1;
        } else {
            $result = 1;
            for ($i = 1; $i <= $num; $i++) {
                $result *= $i;
            }
            return $result;
        }
    }
    if (isset($_GET['number'])) {
        $number = $_GET['number'];
        if (is_numeric($number) && $number >= 0) {
            $factorial = factorial($number);
```

```
        echo "Factorial of $number is $factorial.";
    } else {
        echo "Please enter a non-negative integer.";
    }
}
?>
</body>
</html>
```

Output:

If you enter number 5:

Factorial of 5 is 120

If you enter negative number:

Factorial is not defined for negative numbers.

3. Write PHP scripts that demonstrate fundamentals PHP number triangle.

```
<!DOCTYPE html>
<html>
<head>
<title>Number Triangle</title>
</head>
<body>
<?php
function numberTriangle($n)
{
   for ($i = 1; $i <= $n; $i++) {
      for ($j = 1; $j <= $i; $j++) {
         echo "$j ";
      }
      echo "<br>";
   }
}
?>
<form action="" method="get">
   Enter the number of rows:
   <input type="number" name="rows">
   <input type="submit" value="Generate Triangle">
</form>
<?php
if (isset($_GET['rows'])) {
   $rows = $_GET['rows'];
   if (is_numeric($rows) && $rows > 0) {
      echo "<h3>Number Triangle with $rows rows:</h3>";
```

```
        numberTriangle($rows);
    } else {
        echo "Please enter a positive integer.";
    }
}
?>
</body>
</html>
```

Output:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

4. **Write PHP script that will display grade based on criteria given below using the marks obtained in the Examination.**
   a) **Distinction (70 and above)**
   b) **First Class (60-69)**
   c) **Pass (40-59)**
   d) **Fail (Below 40)**

```
<!DOCTYPE html>
<html>
<head>
<title>Grade Calculator</title>
</head>
<body>
  <form action="" method="post">
    Enter marks obtained:
    <input type="number" name="marks" min="0" max="100">
    <input type="submit" value="Calculate Grade">
  </form>
  <?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if(isset($_POST['marks'])) {
      $marks = $_POST['marks'];
      if(is_numeric($marks) && $marks >= 0 && $marks <= 100) {
        echo "<h3>Grade based on $marks marks:</h3>";
        if ($marks >= 70 && $marks <= 100) {
          echo "Grade: Distinction";
        } elseif ($marks >= 60 && $marks < 70) {
          echo "Grade: First Class";
        } elseif ($marks >= 50 && $marks < 60) {
          echo "Grade: Second Class";
        } elseif ($marks >= 40 && $marks < 50) {
```

```
            echo "Grade: Pass";
        } else {
            echo "Grade: Fail";
        }
    } else {
        echo "Please enter a valid mark between 0 and 100.";
    }
  }
  ?>
</body>
</html>
```

Output:

Grade based on 85 marks:

Grade: Distinction

Grade based on 62 marks:

Grade: First Class

Grade based on 48 marks:

Grade: Pass

Grade based on 30 marks:

Grade: Fail

5. Write a PHP script to demonstrate different String functions.

```
<!DOCTYPE html>
<html>
<head>
<title>String Functions</title>
</head>
<body>
<?php
$string = "Hello World!";
echo "<h3>Original String:</h3>";
echo $string . "<br><br>";
echo "<h3>String Functions:</h3>";
// String Length
echo "1. String Length (strlen): " . strlen($string) . "<br>";
// String to Uppercase
echo "2. String to Uppercase (strtoupper): " . strtoupper($string) . "<br>";
// String to Lowercase
echo "3. String to Lowercase (strtolower): " . strtolower($string) . "<br>";
// Substring
echo "4. Substring (substr): " . substr($string, 6) . "<br>";
// String Replace
echo "5. String Replace (str_replace): " . str_replace("World", "PHP", $string) . "<br>";
// String Reverse
echo "6. String Reverse (strrev): " . strrev($string) . "<br>";
// Trim
$string_with_spaces = "   Hello World!   ";
echo "7. Trim (trim): " . trim($string_with_spaces) . "<br>";
// Word Count
echo "8. Word Count (str_word_count): " . str_word_count($string) . "<br>";
```

```
// String Shuffle
echo "9. String Shuffle (str_shuffle): " . str_shuffle($string) . "<br>";
// String Position
$substring = "World";
echo "10. String Position (strpos): " . strpos($string, $substring) . "<br>";
// String Repeat
echo "11. String Repeat (str_repeat): " . str_repeat($string, 2) . "<br>";
// String Comparison
$string1 = "hello";
$string2 = "HELLO";
echo "12. String Comparison (strcasecmp): " . strcasecmp($string1, $string2) . "<br>";
?>
</body>
</html>
```

Output:

String Functions:

1. String Length (strlen): 12

2. String to Uppercase (strtoupper): HELLO WORLD!

3. String to Lowercase (strtolower): hello world!

4. Substring (substr): World!

5. String Replace (str_replace): Hello PHP!

6. String Reverse (strrev): !dlroW olleH

7. Trim (trim): Hello World!

8. Word Count (str_word_count): 2

9. String Shuffle (str_shuffle): rdl!lHloo Weo

10. String Position (strpos): 6

11. String Repeat (str_repeat): Hello World!Hello World!

12. String Comparison (strcasecmp): 0

6. Write PHP script to demonstrate OOPS concept in PHP.

```
<?php
// Define a class called 'Car'
class Car {
    // Properties (attributes) of the Car class
    public $brand;
    public $model;
    public $color;
    public $speed;
    // Constructor method to initialize object properties
    public function __construct($brand, $model, $color, $speed) {
        $this->brand = $brand;
        $this->model = $model;
        $this->color = $color;
        $this->speed = $speed;
    }
    // Method to display car details
    public function displayDetails() {
        echo "Brand: " . $this->brand . "<br>";
        echo "Model: " . $this->model . "<br>";
        echo "Color: " . $this->color . "<br>";
        echo "Speed: " . $this->speed . " km/h<br>";
    }
    // Method to increase speed
    public function increaseSpeed($increment) {
        $this->speed += $increment;
        echo "Speed increased by $increment km/h. New speed: " . $this->speed . " km/h<br>";
    }
```

```
    // Method to decrease speed
    public function decreaseSpeed($decrement) {
        $this->speed -= $decrement;
        echo "Speed decreased by $decrement km/h. New speed: " . $this->speed . " km/h<br>";
    }
}
// Create objects of the Car class
$car1 = new Car("Toyota", "Corolla", "Red", 100);
$car2 = new Car("Honda", "Civic", "Blue", 120);
// Display car details
echo "<h3>Car 1 Details:</h3>";
$car1->displayDetails();
echo "<br>";
echo "<h3>Car 2 Details:</h3>";
$car2->displayDetails();
echo "<br>";
// Increase speed of car1
echo "<h3>Increasing speed of Car 1:</h3>";
$car1->increaseSpeed(20);
echo "<br>";
// Decrease speed of car2
echo "<h3>Decreasing speed of Car 2:</h3>";
$car2->decreaseSpeed(10);
?>
```

Output:

Car 1 Details:

Brand: Toyota

Model: Corolla

Color: Red

Speed: 100 km/h

Car 2 Details:

Brand: Honda

Model: Civic

Color: Blue

Speed: 120 km/h

Increasing speed of Car 1:

Speed increased by 20 km/h. New speed: 120 km/h

Decreasing speed of Car 2:

Speed decreased by 10 km/h. New speed: 110 km/h

7. Write a PHP script to demonstrate form Data Handling using Get and Post method.

```
<!DOCTYPE html>
<html>
<head>
<title>Form Data Handling</title>
</head>
<body>
<!-- Form using GET method -->
<h3>Form using GET method:</h3>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="get">
   Name: <input type="text" name="name"><br>
   Email: <input type="email" name="email"><br>
   <input type="submit" value="Submit">
</form>
<?php
// Handling form data using GET method
if(isset($_GET['name']) && isset($_GET['email'])) {
   $name = $_GET['name'];
   $email = $_GET['email'];
   echo "<h4>Received using GET method:</h4>";
   echo "Name: $name<br>";
   echo "Email: $email<br>";
}
?>
<hr>
<!-- Form using POST method -->
<h3>Form using POST method:</h3>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
```

```
    Name: <input type="text" name="name"><br>
    Email: <input type="email" name="email"><br>
    <input type="submit" value="Submit">
</form>
<?php
// Handling form data using POST method
if(isset($_POST['name']) && isset($_POST['email'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    echo "<h4>Received using POST method:</h4>";
    echo "Name: $name<br>";
    echo "Email: $email<br>";
}
?>
</body>
</html>
```

Output:

If you fill out the form using the GET method and submit it with name as "John" and email as "john@example.com", the output will be:

Received using GET method: Name: John Email: john@example.com

If you fill out the form using the POST method and submit it with name as "Alice" and email as "alice@example.com", the output will be:

Received using POST method: Name: Alice Email: alice@example.com

8. Design a database in MySQL Create table in database. Store, Update, Delete and Retrieve data from the table. Display the data from the table.

1) Connect to mysql:

```
<?php
$servername = "localhost";
$username = "username"; // Your MySQL username
$password = "password"; // Your MySQL password
$database = "my_database"; // Your MySQL database name

// Create connection
$conn = new mysqli($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
?>
```

2) Insert data into table:

```
<?php
// Retrieve data
$sql = "SELECT * FROM users";
$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["name"]. " - Email: " . $row["email"]. " - Age: " .
$row["age"]. "<br>";
   }
} else {
   echo "0 results";
}
?>
```

3) Retrieve data from table:

```
<?php
// Retrieve data
$sql = "SELECT * FROM users";
$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["name"]. " - Email: " . $row["email"]. " - Age: " .
$row["age"]. "<br>";
    }
} else {
    echo "0 results";
}
?>
```

4) Update data in the table:

```
<?php
// Update data
$sql = "UPDATE users SET age = 40 WHERE id = 1";
if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}
?>
```

5) Delete data from the table:

```
<?php
// Delete data
$sql = "DELETE FROM users WHERE id = 3";
if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;
}
?>
```

6) Close the database connection:

```
<?php
// Close connection
$conn->close();
?>
```

Output:

New record created successfully

id: 1 - Name: John Doe - Email: john@example.com - Age: 30

id: 2 - Name: Alice Smith - Email: alice@example.com - Age: 25

id: 3 - Name: Bob Johnson - Email: bob@example.com - Age: 35

Record updated successfully

Record deleted successfully

9. Write a PHP script to store retrieve and delete cookies on your local machine.

```
<!DOCTYPE html>
<html>
<head>
   <title>Cookie Handling</title>
</head>
<body>
<?php
// Store cookies
setcookie("username", "JohnDoe", time() + (86400 * 30), "/"); // Cookie for 30 days
setcookie("language", "PHP", time() + (86400 * 30), "/");
// Retrieve cookies
echo "<h3>Retrieved Cookies:</h3>";
if(isset($_COOKIE["username"])) {
   echo "Username: " . $_COOKIE["username"] . "<br>";
} else {
   echo "Username cookie not set.<br>";
}
if(isset($_COOKIE["language"])) {
   echo "Language: " . $_COOKIE["language"] . "<br>";
} else {
   echo "Language cookie not set.<br>";
}
// Delete cookies
echo "<h3>Deleted Cookies:</h3>";
if(isset($_COOKIE["username"])) {
   setcookie("username", "", time() - 3600, "/"); // Set cookie expiration time to past
```

```
    echo "Username cookie deleted.<br>";
}
if(isset($_COOKIE["language"])) {
    setcookie("language", "", time() - 3600, "/");
    echo "Language cookie deleted.<br>";
}
?>
</body>
</html>
```

Output:

Retrieved Cookies:

Username: JohnDoe

Language: PHP

Deleted Cookies:

Username cookie deleted.

Language cookie deleted.

10. Write a PHP script to store, retrieve and delete data using session variables.

```
<?php
// Start the session
session_start();
// Store data in session variables
$_SESSION["username"] = "JohnDoe";
$_SESSION["email"] = "john@example.com";
// Retrieve data from session variables
echo "<h3>Retrieved Data:</h3>";
if(isset($_SESSION["username"])) {
    echo "Username: " . $_SESSION["username"] . "<br>";
} else {
    echo "Username not set.<br>";
}
if(isset($_SESSION["email"])) {
    echo "Email: " . $_SESSION["email"] . "<br>";
} else {
    echo "Email not set.<br>";
}

// Delete data from session variables
echo "<h3>Deleted Data:</h3>";
if(isset($_SESSION["username"])) {
    unset($_SESSION["username"]);
    echo "Username deleted.<br>";
}
if(isset($_SESSION["email"])) {
```

```
    unset($_SESSION["email"]);
    echo "Email deleted.<br>";
}
// End the session
session_destroy();
?>
```

Output:

Retrieved Data:

Username: JohnDoe

Email: john@example.com

Deleted Data:

Username deleted.

Email deleted.