
V2V EDTECH LLP
Online Coaching at an Affordable Price.
OUR SERVICES:
• Diploma in All Branches, All Subjects
• Degree in All Branches, All Subjects
• BSCIT / CS
• Professional Courses
+91 93260 50669
V2V EdTech LLP
v2vedtech.com
v2vedtech

UNIT 1 Expressions and Control statements in PHP

Q. 1 Describe advantages of PHP. (S-22) 2 Marks
List any four advantages of PHP. (S-24) 2 Marks

Ans : following are the advantages of php :

- Open Source : PHP is open source and free of cost, which helps developers to install it quickly and readily available for use. There are a lot of PHP frameworks and developer can choose any of the frameworks to work. All the features and tools will be provided to the developer for that framework easily. Open Source means you never need to rely on the manufacturer to release the next version if something doesn't work or pay for expensive upgrades.
- Platform Independent: PHP is mainly supported by all the operating systems like Windows, Unix, Linux etc. The PHP based developed web applications can be easily run on any platform. It can be integrated with other programming language and database easily and there is no requirement of re-development. It helps in saving a lot of effort and cost.
- simple and Easy: PHP is simple and easy to learn and code. The command functions of PHP can easily learn and understood. The syntax is simple and flexible to use.
- Database: PHP is easily connected with the database and make the connection securely with databases. It has a built-in module that is used to connect to the database easily. Multiple databases can be integrated with PHP.
- Fast: PHP is known as the fastest Programming language as compared to another. PHP applications can be easily loaded over the slow Internet and data speed.
- Support: This advantage of PHP has great online support and community, which helps the new developers to help in writing the code and developing the web applications. Compatible with servers like IIS and APACHE.
- Security: PHP frameworks built-in feature and tools make it easier to protect the web applications from the outer attacks and security threats.

- Maintenance : PHP framework is mainly used to make the web application development easier and maintain the code automatically. Low development and maintenance cost with very high performance and reliability
- Fast Performance: Scripts written in PHP usually execute faster than those written in other scripting languages like ASP.NET or JSP.
- Vast Community: Since PHP is supported by the worldwide community, finding help or documentation for PHP online is extremely easy. PHP is so popular that it's quite easy for you to get support on PHP.
- Scripting language: A scripting language or script language is a programming language that supports scripts. If code of programming language can embed with other language or integrate with other language or script called scripting language. PHP is Scripting language because we can embed PHP code into HTML. PHP is server-side language because PHP requires server to run a code. Code of PHP gets executed on server and result of execution is return to the browser, that's why PHP is called script language and server-side language.

Q.2 Write syntax of PHP. (S-22) 2 Marks

Ans: A PHP script starts with the tag. The PHP delimiter in the following example simply tells the PHP engine to treat the enclosed code block as PHP code, rather than simple HTML.
On servers with shorthand support enabled, you can start a scripting block with.
Syntax: <?php

?>

Example: Each code line in PHP must end with a semicolon.
The semicolon is a separator and is used to distinguish one set of instructions from another.
There are two basic statements to output text with PHP: echo and print.
In the example above we have used the echo statement to output the text "Hello World".

Q.3 Write down rules for declaring PHP variables. (S-22) 4 Marks

Ans : Rules for PHP variables:

- A variable starts with the $ sign, followed by the name of the variable.
- A variable name must start with a letter or the underscore character.
- A variable name should not contain spaces. If a variable name is more than one word, it should be separated with an underscore ($first_name), or with capitalisation ($firstName).
- Variables used before they are assigned have default values.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters (A-Z, a-z) and underscores.
- Variable names are case-sensitive ($name and $NAME are two different variables)

- You able to use variable over and over again in your PHP script after declaring it.
- Variables can, but do not need, to be declared before assignment. PHP automatically converts the variable to the correct data type, depending on its value.
- Variables in PHP do not have intrinsic types - a variable does not know in advance whether it will be used to store a number or a string of character.

Q.4 Explain different loops in PHP with example. (S-22) 6 Marks

Ans : Loops in PHP are used to execute the same block of code a specified number of times. PHP supports the following four loop types:

- **for** − loops through a block of code a specified number of times.
- **while** − loops through a block of code if and as long as a specified condition is true.
- **do...while** − loops through a block of code once, and then repeats the loop as long as a special condition is true.
- **foreach** − loops through a block of code for each element in an array.

The do...while statement will execute a block of code at least once - it then will repeat the loop as long as a condition is true.
The following example will increment the value of i at least once, and it will continue incrementing the variable i as long as it has a value of less than 10.
Example :

```
<html>
  <body>
      <?php
      $i = 0;
      $num = 0;
      do {
        $i++;
      }
       while( $i < 10 );
      echo ("Loop stopped at i = $i" );
    ?>

  </body>
</html>
```

This will produce the following result −
Loop stopped at i = 10

While Loop-The while loop - Loops through a block of code as long as the specified condition is true.

```
$i = 1;
while ($i < 6) {
  echo $i;
  $i++;
}
```

The while loop does not run a specific number of times, but checks after each iteration if the condition is still true.
The condition does not have to be a counter, it could be the status of an operation or any condition that evaluates to either true or false.

For loop- The `for` loop - Loops through a block of code a specified number of times.
The for loop is used when you know how many times the script should run.

Syntax-

```
for (expression1, expression2, expression3) {
  // code block
}
```

This is how it works:
expression1 is evaluated once
expression2 is evaluated before each iteration
expression3 is evaluated after each iteration
example-

```
<?php
for ($x = 0; $x <= 10; $x++) {
  echo "The number is: $x <br>";
}
?>
```

- The first expression, $x = 0;, is evaluated once and sets a counter to 0.
- The second expression, $x <= 10;, is evaluated before each iteration, and the code block is only executed if this expression evaluates to true. In this example the expression is true as long as $x is less than, or equal to, 10.
- The third expression, $x++;, is evaluated after each iteration, and in this example, the expression increases the value of $x by one at each iteration.

For each loop- The foreach loop - Loops through a block of code for each element in an array or each property in an object.The most common use of the foreach loop, is to loop through the items of an array.

```
$colors = array("red", "green", "blue", "yellow");
```

```
foreach ($colors as $x)
{
 echo "$x <br>";
}
$members = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
foreach ($members as $x => $y)
{
  echo "$x : $y <br>";
}
```

Q.5 List any four data types of PHP. (W-22) 2 Marks

Ans: Data types are used to hold different types of data or values.

PHP supports 8 primitive data types that can be categorized in 3 types:

- Scalar Type- Integer, float, string, Boolean
- Compound Type-array, object
- Special type-resource, null

Q.6 Define Array. State its example. (W-22) 2 Marks

Ans: An array in PHP is an ordered map. A map is a type that associates values with keys.

Arrays in PHP is a type of data structure that allows us to store multiple elements of similar data type under a single variable thereby saving us the effort of creating a different variable for every data.

An array is created using an array () function in PHP.

```
<?php
$course[0]="Computer Engg.";
$course[1]="Information  Tech.";
$course[2]="Electronics and Telecomm.";
// Accessing the elements directly
echo $course[2], "<BR>";
echo $course[0], "<BR>";
echo $course[1], "<BR>";
?>
```

Q.7 Write syntax of for each loop. (W-22) 2 Marks

Ans:

```
foreach (array_element as value)
            {
              //code to be executed
            }
```

Q.8 Explain the use of break and continue statements. (W-22) 4 Marks

Ans: The PHP break keyword is used to terminate the execution of a loop prematurely. The break statement is situated inside the statement block. It gives you full control and whenever you want to exit from the loop you can come out. After coming out of a loop immediate statement to the loop will be executed.

```
<?php
      $i = 0;
      while( $i < 10)
{
        $i++;
        if( $i == 3 )
       break;
      }
      echo ("Loop stopped at i = $i" );
   ?>
```

The PHP continue keyword is used to halt the current iteration of a loop but it does not terminate the loop.
Just like the break statement the continue statement is situated inside the statement block containing the code that the loop executes, preceded by a conditional test. For the pass encountering continue statement, rest of the loop code is skipped and next pass starts

```
<?php
      $array = array( 1, 2, 3, 4, 5);
      foreach( $array as $value )
     {
        if( $value == 3 )
continue;
        echo "Value is $value <br />";
      }
   ?>
```

Q.9 Describe the syntax of if-else control statement with example in PHP. (W-22) 4 Marks

Ans 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!";
     ?>
```

Q.10 Write a PHP program to display numbers from 1-10 in a sequence using for loop. (W-22) 4 Marks

Ans:

```
<?php
        $a = 1;
        for( $a = ; $a<=10; $a++ )
      {
          echo $a;
        }
        ?>
```

Q.11 State the advantages of PHP (any four). (S-23) 2 Marks

Ans :Advantages of php :

1) Open Source
2) Platform Independent
3) Simple and Easy
4) Database
5) Fast
6) Support
7) Security
8) Maintenance

9) Fast Performance
10) Vast Community
11) Scripting language

Q.12 State the use of $ sign in PHP. (S-23) 2 Marks
Ans: The $var (single dollar) is a normal variable with the name var that stores any value like string, integer, float, etc.
Example: $x=" Hello";
In PHP, a variable starts with the $ sign, followed by the name of the variable
The assignment operator (=) used to assign value to a variable.
In PHP variable can be declared as: $var_name = value;

Q.13 Write a program using do-while loop. (S-23) 4 Marks
Ans : The do...while statement will execute a block of code at least once - it then will repeat the loop as long as a condition is true.
The following example will increment the value of i at least once, and it will continue incrementing the variable i as long as it has a value of less than 10.
Example :

```
<html>
   <body>
        <?php
        $i = 0;
        $num = 0;
        do {
          $i++;
        }
         while( $i < 10 );
        echo ("Loop stopped at i = $i" );
      ?>

   </body>
</html>
```

This will produce the following result −
Loop stopped at i = 10

Q.14 Implement any three datatypes used in PHP with illustration. (S-23) 6 Marks
Ans:Datatyapes are Array,integer

Array: An array stores multiple values in one single variable and each value is identify by position (zero is the first p position). The array is a collection of heterogeneous (dissimilar) data types. PHP is a loosely typed language that's why we can store any type of values in arrays.

Syntax : Variable_name = array (element1, element2, element3, element4. ...)

```
$cars = array("Volvo", "BMW", "Toyota");
foreach ($cars as $x) {
  echo "$x <br>";
}
```

Integer:

- integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.
- Rules for integers:
- An integer must have at least one digit
- An integer must not have a decimal point
- An integer can be either positive or negative
- Integers can be specified in: decimal (base 10), hexadecimal (base 16), octal (base 8), or binary (base 2) notation
- In the following example $x is an integer. The PHP var_dump() function returns the data type and value:
- Example :

```
$x = 5985;
var_dump($x);
int(5985)
```

Boolean :

- A Boolean represents two possible states: TRUE or FALSE.
- Example :$x = true;
  var_dump($x);

  Output is bool(true)

Float :

- A float (floating point number) is a number with a decimal point or a number in exponential form.
- In the following example $x is a float. The PHP var_dump() function returns the data type and value:
- Example :

```
$x = 10.365;
var_dump($x)
float(10.365)
```

Q.15 State any 2 differences between for and for each. (W-23) 2 Marks

Ans: Difference is as follows

for loop	foreach loop
The iteration is clearly visible. The block of code is repeated if the condition is met, or the counter meets a specific value.	The iteration is hidden. The block of code is repeated until iterating over the array is completed.
Good performance.	Better performance.
The stop condition is specified easily.	The stop condition must be specified.
Upon working with collections, it needs the usage of the count() function.	It can simply work without the usage of the count() method.

Q.16 List loop control structures. Explain any one loop control structure. (W-23) 4 Marks

Ans : Loop control structures are while,do...while ,for loop and for each loop

- Do...while loop :
- The do... while statement will execute a code block at least once, it will repeat the loop if a condition is true.
- The following example will increment the value of i at least once, and it will continue incrementing the variable i as long as it has a value of less than 10.
- Example :

```
<html>
   <body>
      <?php
      $i = 0;
      $num = 0;

      do {
        $i++;
      }
            while( $i < 10 );
      echo ("Loop stopped at i = $i" );
   ?>

  </body>
</html>
```

This will produce the following result −

Loop stopped at i = 10

Q.17 Explain the following terms: i) Variables ii) Datatypes iii) Constant iv) Operators
(W-23) 4 Marks
Ans:

i) Variables:

- Variables are "containers" for storing information.
- A variable can have a short name (like $x and $y) or a more descriptive name ($age, $carname, $total_volume).
- Rules for PHP variables:
  - A variable starts with the $ sign, followed by the name of the variable
  - A variable name must start with a letter or the underscore character
  - A variable name cannot start with a number
  - A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
  - Variable names are case-sensitive ($age and $AGE are two different variables)

iii) Constant

- Constants are like variables, except that once they are defined they cannot be changed or undefined.
- A constant is an identifier (name) for a simple value. The value cannot be changed during the script.
- A valid constant name starts with a letter or underscore (no $ sign before the constant name).
- To create a constant, use the define() function.
- Parameters:
  - *name*: Specifies the name of the constant
  - *value*: Specifies the value of the constant
  - *case-insensitive*: Specifies whether the constant name should be case-insensitive. Default is false. Note: Defining case-insensitive constants was deprecated in PHP 7.3. PHP 8.0 accepts only false, the value true will produce a warning.

```
define("GREETING", "Welcome to W3Schools.com!");
echo GREETING;
```

Welcome to W3Schools.com!

ii) Datatypes-

- Variables can store data of different types, and different data types can do different things.
- PHP supports the following data types:
  - String
  - Integer
  - Float (floating point numbers - also called double)

- Boolean
- Array
- Object
- NULL
- Resource

iv) operators-

- Operators are used to perform operations on variables and values.
- PHP divides the operators in the following groups:
  - Arithmetic operators
  - Assignment operators
  - Comparison operators
  - Increment/Decrement operators
  - Logical operators
  - String operators
  - Array operators
  - Conditional assignment operators

Q.18 State different data types in php. W-24

Ans- Variables can store data of different types, and different data types can do different things.
PHP supports the following data types:

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

Q.19 Explain bitwise operators in php . W-24

Ans- The Bitwise operators is used to perform bit-level operations on the operands. The operators are first converted to bit-level and then calculation is performed on the operands. The mathematical operations such as addition , subtraction , multiplication etc. can be performed at bit-level for faster processing. In PHP, the operators that works at bit level are:

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

$First & $Second This will return another number whose bits are
set if both the bit of first and second are set.
Example:

Input: $First = 5, $Second = 3

Output: The bitwise & of both these value will be 1.

Explanation:
Binary representation of 5 is 0101 and 3 is 0011.
Therefore their bitwise & will be 0001 (i.e. set
if both first and second have their bit set.)
| (Bitwise OR) : This is also binary operator i.e. works on two operand. Bitwise OR operator takes two numbers as operands and does OR on every bit of two numbers. The result of OR is 1 any of the two bits is 1.
Syntax:

$First | $Second

This will return another number whose bits are
set if either the bit of first or second are set.
Example:

Input: First = 5, Second = 3

Output: The bitwise | of both these value will be 7.

Explanation:
Binary representation of 5 is 0101 and 3 is 0011.
Therefore their bitwise | will be 0111 (i.e. set
if either first or second have their bit set.)
^ (Bitwise XOR) : This is also binary operator i.e. works on two operand. This is also known as Exclusive OR operator. Bitwise XOR takes two numbers as operands and does XOR on every bit of two numbers. The result of XOR is 1 if the two bits are different.
Syntax:

$First ^ $Second

This will return another number whose bits are
set if one of the bit in first or second is
set but not both.
Example:

Input: First = 5, Second = 3

Output: The bitwise ^ of both these value will be 6.

Explanation:
Binary representation of 5 is 0101 and 3 is 0011.
Therefore their bitwise ^ will be 0110 (i.e. set
if either first or second have their bit set but
not both.)
~ (Bitwise NOT) : This is a unary operator i.e. works on only one operand. Bitwise NOT operator takes one number and inverts all bits of it.
Syntax:

~$number

This will invert all the bits of $number.

Example:

Input: number = 5

Output: The bitwise '~' of this number will be -6.

Explanation:
Binary representation of 5 is 0101. Therefore the
bitwise ~ of this will be 1010 (inverts all the
bits of the input number)
<< (Bitwise Left Shift) : This is a binary operator i.e. works on two operand. Bitwise Left Shift operator takes two numbers, left shifts the bits of the first operand, the second operand decides the number of places to shift.
Syntax:

$First << $Second

This will shift the bits of $First towards the
left. $Second decides the number of time the
bits will be shifted.
Example:

Input: First = 5, Second = 1

Output: The bitwise << of both these value will be 10.

Explanation:
Binary representation of 5 is 0101 . Therefore,
bitwise << will shift the bits of 5 one times
towards the left (i.e. 01010)
Note: Bitwise left shift with one bit is equivalent to multiplication with 2.

>> (Bitwise Right Shift) : This is also binary operator i.e. works on two operand. Bitwise Right Shift operator takes two numbers, right shifts the bits of the first operand, the second operand decides the number of places to shift.
Syntax:

$First >> $Second

This will shift the bits of $First towards the
right. $Second decides the number of time the
bits will be shifted.
Example:

Input: First = 5, Second = 1

Output: The bitwise >> of both these value will be 2.

Explanation:
Binary representation of 5 is 0101 . Therefore,
bitwise >> will shift the bits of 5 one times
towards the right(i.e. 010)
Note: Bitwise right shift with one bit is equivalent to division with 2.

Below is the implementation of Bitwise Operators in PHP:

```
<?php
    // PHP code to demonstrate Bitwise Operator.

    // Bitwise AND
    $First = 5;
    $second = 3;
    $answer = $First & $second;

    print_r("Bitwise & of 5 and 3 is $answer");

    print_r("\n");
```

```
    // Bitwise OR
    $answer = $First | $second;
    print_r("Bitwise | of 5 and 3 is $answer");

    print_r("\n");

    // Bitwise XOR
    $answer = $First ^ $second;
    print_r("Bitwise ^ of 5 and 3 is $answer");

    print_r("\n");

    // Bitwise NOT
    $answer = ~$First;
    print_r("Bitwise ~ of 5 is $answer");

    print_r("\n");

    // Bitwise Left shift
    $second = 1;
    $answer = $First << $second;
    print_r("5 << 1 will be $answer");

    print_r("\n");

    // Bitwise Right shift
    $answer = $First >> $second;
    print_r("5 >> 1 will be $answer");

    print_r("\n");
?>
```

Output:

Bitwise & of 5 and 3 is 1
Bitwise | of 5 and 3 is 7
Bitwise ^ of 5 **and 3 is 6**
Bitwise ~ of 5 is -6
5 << 1 will be 10
5 >> 1 will be 2

Q.20 Write a php program to find largest between to numbers w-24
Ans-
The simplest use case of the ternary operator is to compare two numbers and return the largest one.

```
<?php

$num1 = 10;
$num2 = 20;

$largest = ($num1 > $num2) ? $num1 : $num2;

echo "Largest Number: $largest";

?>
```

Output
Largest Number: 20

UNIT 2 Arrays, functions and Graphics

Q.1 What is array? How to store data in array? (S-22) 2 Marks

Ans:

- Array is a special variable that we use to store or hold more than one value in a single variable without having to create more variables to store those values.
- To create an array in PHP, we use the array function array().By default, an array of any variable starts with the 0 index.
- whenever you want to call the first value of an array you start with 0 then the next is 1...and so on.
- There are different types of arrays in PHP. They are: 1) Numeric/Indexed Arrays 2)Associative Arrays 3) Multidimensional Arrays
- Example –

```
<?php
$course = array(0 => "Computer Engg.",1 => "Information Tech.", 2 => "Electronics and Telecomm.");
echo $course[1];
?>
```

Q.2 Write a program to create associative array in PHP. (S-22) 4 Marks

Ans : Associative array is a type of array where the key has its own value. In an associative array, we make use of key and value.Keys are descriptive captions of the array element used to access the value of the array. And value is the value assigned to the array element.

There are situations where you shouldn't use the numeric/indexed array, such as:

- When you want to store the age of different students along with their names.
- When you want to record the salaries of your employees.
- When you want to store the score of a student in different subjects.

Suppose we want to assign ages to a group of high school students with their names.

We can use the Associative array method to get it done. For example:

```
<?php
$course = array("CO"=>549, "IF"=>450,"EJ"=>100);
echo "course['CO']=", $course["CO"],"<br>";
echo "course['IF']=", $course["IF"],"<br>";
```

```
echo "course['EJ']=", $course["EJ"],"<br>";
?>
```

Q.3 Define function. How to define user defined function in PHP? Give example. (S-22) 4 Marks

Ans

- PHP functions are like other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.
- They are built-in functions, but PHP gives you option to create your own functions as well. A function will be executed by a call to the function. You may call a function from anywhere within a page.
- **Syntax and example :**

Example 1

```
function functionName()
{
  code to be executed;
}
```

```
<?php
/* Defining a PHP Function */
function writeMessage()
{
        echo "Welcome to PHP world!";
}
/* Calling a PHP Function */
      writeMessage();
?>
```

Example 2

```
<?php
       function addfunc($num1, $num2)
      {
         $sum = $num1 + $num2;
```

```
        echo "Sum of the two numbers is : $sum";
      }
      addfunc(50, 20);
    ?>
```

Q.4 Write PHP script to sort any five numbers using array function. (S-22) 4 Marks

Ans :

```
<?php
$num = array(40, 61, 2, 22, 13);
echo "Before Sorting:<br>";
$arrlen= count($num);
for($x = 0; $x < $arrlen; $x++)
  {
   echo $num[$x];
   echo "<br>";
  }
sort($num);
echo "After Sorting in Ascending order:<br>";
$arrlen= count($num);
for($x = 0; $x < $arrlen; $x++)
  {
   echo $num[$x];
   echo "<br>";
  }
echo "After Sorting in Descending order:<br>";
rsort($num);
$arrlen= count($num);
for($x = 0; $x < $arrlen; $x++)
  {
   echo $num[$x];
   echo "<br>";
  }
?>
```

Q.6 Explain any four string functions in PHP with example. (S-22) 6 Marks

Ans : following are strings functions below:

- Word Count -
  - The PHP str_word_count() function counts the number of words in a string.
  - Example - Count the number of word in the string "Hello world!":

    echo str_word_count("Hello world!"); output is 2
- Strpos ( ) -
  - The PHP strpos() function searches for a specific text within a string.
  - If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.
  - Example - Search for the text "world" in the string "Hello world!":

    echo strpos("Hello world!", "world");
- str_replace( ) -
  - Replaces some characters in a string (case-sensitive)
  - str_replace(string tobe replaced, text, string)

```
<?php  echo str_replace("Clock","Click","Click and Clock"); ?>
```

Output: Click and Click

- strcmp( )-
  - Compare two strings (case-sensitive). If this function returns 0, the two strings are equal.If this function returns any negative or positive numbers, the two strings are not equal.
  - Strcmp(String1, String2)

```
<?php echo strcmp(" world!","Hello PHP!"); ?>
```

output is 40

Q.7 Explain Indexed array and associative arrays with suitable example. (W-22) 6 Marks
Ans :

- 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.",
    2 => "Electronics and Telecomm."); echo $course[1]; ?>

- 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 use descriptive names for id keys.
  - Syntax : <?php $variable_name['key_name'] = value;
    $variable_name = array('keyname' => value); ?>

Q.8 Describe user defined function with example. (W-22) 4 Marks
Ans :

- PHP functions are like other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.
- They are built-in functions, but PHP gives you option to create your own functions as well. A function will be executed by a call to the function. You may call a function from anywhere within a page.
- **Syntax and example :**

Example 1

```
function functionName()
{
  code to be executed;
```

```
}

<?php
/* Defining a PHP Function */
function writeMessage()
{
        echo "Welcome to PHP world!";
}
/* Calling a PHP Function */
      writeMessage();
?>
```

Example 2

```
<?php
       function addfunc($num1, $num2)
      {
         $sum = $num1 + $num2;
         echo "Sum of the two numbers is : $sum";
       }
       addfunc(50, 20);
    ?>
```

Q.9 Write a PHP program to (i) Calculate length of string (ii) Count number of words in the string(W-22) 6 Marks

Ans : (i) Calculate length of string - Returns the length of a string

<?php echo strlen("Welcome to PHP"); ?> output is 14

(ii) Count number of words in the string - Count the number of words in a string

<?php echo str_word_count("Welcome to PHP world!"); ?> output is 4

Q.10 State the use of strlen() &strrev(). (S-23) 6 Marks

Ans :

- **strlen( ) -**
  - Returns the length of a string.
  - Synatx - strlen(String)

- Example - <?php echo strlen("Welcome to PHP"); ?

- **strrev():**
  - Reverses a string
  - Synatx - strrev(String)
  - Example- <?php echo strrev("Information Technology"); ?>

Q.11 Explain associative and multi-dimensional arrays. (S-23) 6 Marks

Ans : Associative arrays -

- 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 differs from numeric array in the sense that associative arrays use descriptive names for id keys.

Syntax :

<?php $variable_name['key_name'] = value; $variable_name = array('keyname' => value); ?>

```
<?php

$capital = array("Mumbai" => "Maharashtra","Goa" => "Panaji", "Bihar" => "Patna");

print_r($capital);

echo"<br>";

echo "Mumbai is a capital of  " .$capital ["Mumbai"];

?>
```

By default, array starts with index 0. What if you wanted to start with index 1 instead? You could use the PHP =>operator like this:

```
$course = array ( 1 => "Computer Engg.", 2 => "Information Tech.", 3 => "Electronics and Telecomm.");
```

The => operator create key/value pairs in arrays-the item on the left of the operator => is the key and the item on the right is the value.

Multidimesional arrays -

- These are arrays that contain other nested arrays.
- An array which contains single or multiple arrays within it and can be accessed via multiple indices.

- We can create one dimensional and two-dimensional array using multidimensional arrays.
- The advantage of multidimensional arrays is that they allow us to group related data together.

Syntax:

array (array (elements...), array (elements...),...)

```
<?php
// Defining a multidimensional array
$person = array(array("name" => "Yogita K",  "mob" => "5689741523","email" => "yogi_k@gmail.com", ),   array( ‘name" => "Manisha P.", "mob" => "2584369721", "email" => "manisha_p@gmail.com", ),
  array( "name" => "Vijay Patil", "mob" => "9875147536", "email" => "Vijay_p@gmail.com", ) );
// Accessing elements
echo "manisha P's email-id is: " . $person[1]["email"], "<br>";
echo "Vijay Patil's mobile no: " . $person[2]["mob"];
?>
<?php
$mobile = array
  (
  array("LG",20,18),  array("sony",30,13),  array("Redme",10,2),  array("Samsung",40,15)  );
echo $mobile[0][0].": In stock: ".$mobile[0][1].", sold: ".$mobile[0][2].".<br>";
echo $mobile[1][0].": In stock: ".$mobile[1][1].", sold: ".$mobile[1][2].".<br>";
echo $mobile[2][0].": In stock: ".$mobile[2][1].", sold: ".$mobile[2][2].".<br>";
echo $mobile[3][0].": In stock: ".$mobile[3][1].", sold: ".$mobile[3][2].".<br>";
?>
```

Q.12 Differentiate between implode and explode functions. (S-23) 4 Marks

Ans : Differentiate between implode and explode function as follows

implode() function	explode() function
It joins an array of elements into strings.	It splits the array based on the separator.
The input type is an array.	The input type is a string.
The output type is a string.	The output type is an array.

It uses the delimiter for string concatenation.	It uses the delimiter for splitting the string.
The syntax of implode() function is implode($delimiter, $array)	The syntax of explode() function is explode($delimiter, $string, $limit)
The output example is ‘“ Car”,” Truck”,” Bus”’.	The output example is ‘Array ([0] => Car [1] => Truck [2] => Bus)

Q.13 State user defined function and explain with example. (S-23) 4 Marks

Ans : PHP functions are similar to other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.

- They are built-in functions, but PHP gives you option to create your own functions as well. A function will be executed by a call to the function. You may call a function from anywhere within a page**.**
- function functionName()

{

code to be executed;

}

Example -

```
<?php
/* Defining a PHP Function */
function writeMessage()
{
        echo "Welcome to PHP world!";
}
/* Calling a PHP Function */
      writeMessage();
?>
 <?php
      function addfunc($num1, $num2)
      {
        $sum = $num1 + $num2;
        echo "Sum of the two numbers is : $sum";
```

```
    }
    addfunc(50, 20);
?>
```

Q.14 List different types of arrays. (W-23) 2 Marks

Ans : Types of arrays -

1) Indexed or Numeric Arrays
2) Associative Arrays
3) Multidimensional Arrays

Q.15 Explain functions. List its types. (W-23) 4 Marks

Ans :

- PHP functions are like other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.
- They are built-in functions, but PHP gives you option to create your own functions as well. A function will be executed by a call to the function. You may call a function from anywhere within a page.
- Types of functions are
  - Variable function
  - PHP Functions with Parameters
  - PHP Functions returning value
  - Variable Function
  - Anonymous or Lambda Function

Q.16 Explain the operations on string i) strrev() ii)strpos() (W-23) 6 Marks

Ans :

i) strrev( )-

- It Reverses a string.
- Example - <?php echo strrev("Information Technology"); ?>
- Output: ygolonhceT noitamrofnI

ii) strpos( )-

- Returns the position of the first occurrence of a string inside another string (case-sensitive).
- Synatx - strpos(String, text)
- Example - <?php echo strpos("PHP contains for loop, for each and while loop","loop"); ?> Output: 17
-

Q.17 List string function in PHP. Explain any two. (W-23) 6 Marks

Ans : string functions are strrev,strlen,strpos, str_word_count( ), str_replace(), ucwords(), strtolower(), strcmp()

i)strrev( )-

- It Reverses a string.
- Example - <?php echo strrev("Information Technology"); ?>
- Output: ygolonhceT noitamrofnI

ii) strpos( )-

- Returns the position of the first occurrence of a string inside another string (case-sensitive).
- Synatx - strpos(String, text)
- Example - <?php echo strpos("PHP contains for loop, for each and while loop","loop"); ?>
- Output: 17

Q.18 Develop a PHP program without using string functions: (W-23) 6 Marks

Ans : i) To calculate length of string.

```
<?php
$s='string';
$i=0;
while($s[$i]!='')
{
$i++;
}
print $i;
?>
```

ii) To count the number of words in string.

```
<?php
// PHP program to count number
// of words in a string
```

```
// Function to count the words
function get_num_of_words($string)
{
    $string = preg_replace('/\s+/', ' ', trim($string));
    $words = explode(" ", $string);
    return count($words);
}
 $str = "  Geeks  for    Geeks  ";
 // Function call
$len = get_num_of_words($str);
  // Printing the result
echo $len;
?>
```

UNIT- 3 OOP Concepts in PHP

Q.1. List types of inheritance. (S-22) 2 Marks

Ans. Types of Inheritance are:

- Single Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance.
- Multiple Inheritance

Q.2. Define Introspection and explain it with suitable example. (S-22, W-22, S-23,S-24) 4 Marks

Ans. Introspection:

Introspection in PHP offers the useful ability to examine an object's characteristics, such as its name, parent class (if any) properties, classes, interfaces, and methods.

In-built functions in PHP Introspection:

Function	Description
class_exists()	Checks whether a class has been defined.
et_class()	Returns the class name of an object.

get parent_class()	Returns the class name of a Return object's parent class.
is_subclass_of()	Checks whether an object has a given parent class.
get_declared_classes()	Returns a list of all declared classes.
get_class_methods()	Returns the names of the class methods.
get_class_vars()	Returns the default properties of a class
interface_exists()	Checks whether the interface is defined.
method_exists()	Checks whether an object defines a method.

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
```

Q.3. Explain method overloading with example. (S-22,W-24) 4 Marks

Ans-

In OOP, Method overloading is a feature that allows creating several methods with a similar name to perform different tasks from one another depending on the input parameters it accepts as arguments.

It is called static polymorphic i.e method overloading.

In the case of PHP, we have to utilize PHP's magic methods __call() to achieve method overloading.

function __call():

Syntax:

function __call(string $function_name, array $arguments) { // block of code}

If a class execute __call(), then if an object of that class is called with a method that doesn't exist then __call() is called instead of that method.

Example:

```
>?php
class Shape {
    const PI = 3.142 ;
    function _call($name,$arg){
      if($name == 'area')
        switch(count($arg)){
          case 0 : return 0 ;
          case 1 : return self::PI * $arg[0] ;
```

```
            case 2 : return $arg[0] * $arg[1];
          }
      }
    }
    $circle = new Shape();
    echo $circle->area(3);
    $rect = new Shape();
    echo $rect->area(8,6);
?>
```

OUTPUT:
9.426
48
In the above example, area () method is created dynamically and executed with the help of magic method __call() and its behavior changes according to the passed number of parameters as object.

Q.4. Write PHP program for cloning of an object. (S-22, W-22, S-23) 4 Marks
Ans-

```
<?php
class   student{
public   $name;
public $course;
public function __construct($n){
$this->name= $n;
}
public function setCourse(course $c){
$this->course =$c;
}
}
class course{
public $cname;
public function __construct($cn){
$this->cname = $cn;
```

```
}
}
$student1 = new student('sneha');
$course1 = new course ('php');
$student1->setCourse($course1);
$student2 = clone $student1;
$student2->name = "Sheetal";
$student2->course->cname ="python";
echo "<pre>";
print_r($student1);
echo "<pre>";
echo "<pre>";
print_r($student2);
echo "<pre>";
?>
```

OUTPUT:

```
student Object
(
    [name] => sneha
    [course] => course Object
        (
            [cname] => python
        )

)

student Object
(
    [name] => Sheetal
    [course] => course Object
        (
            [cname] => python
        )

)
```

Q.6. Create a class as "Percentage" with two properties length & width. Calculate area of rectangle for two objects. (S-22) 4 Marks

Ans-

```
<?php
class Rectangle
{
// Declare properties
```

```
public $length ;
public $width ;
 // define parameterised constructor
function _construct($l,$w){
 $this->length = $l;
 $this->width =$w;
}
// Method to get the area
public function getArea()
{
return ($this->length * $this->width);
}
}
// Create multiple objects from the Rectangle class
$rect1 = new Rectangle(2,5);
$rect2 = new Rectangle(3,8);
// Call the methods of both the objects
echo "Area of Rectangle_1 :" . $rect1->getArea() . "<br>";
echo "Area of Rectangle_2:" .$rect2->getArea(). "<br>";

?>
```

OUTPUT:

Area of Rectangle_1 :10
Area of Rectangle_2:24

Q.7 (i) What is inheritance ? (ii) Write update operation on table data. (S-22,S-24) 6 Marks

Ans:

(I) 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(“Sneha”);
$s1->display();
?>
```

OUTPUT:

```
College name=Vidyalankar Polytechnic
College code=568
Student name=Sneha
```

(II) **Update query:**

pdate data :

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "ifdept";
$conn = new mysqli($servername, $username, $password,
$dbname);
if ($conn->connect_error)
{
die("Connection failed: " . $conn->connect_error);
}
$sql = "UPDATE student SET rollno=4 WHERE
name='abc'";
if ($conn->query($sql) === TRUE)
{
echo "Record updated successfully";
} else
{
echo "Error updating record: " . $conn->error;
}
$conn->close();
?>
```

Q.8 State the role of constructor. (W-22) 2 Marks

Ans.

- Constructor is a special member function of a class.
- It is used to initialize an object automatically when it is created.
- Syntax:

Function __construct(args list){
// block of code.
}

Q.9. Write syntax to create class and object in PHP. (W-22) 4 Marks

Ans.

Syntax to create class:

```
class ClassName{
// data member1;
----------
// data memberN;
        //member function1 ();
----------
        //member functionN ()
        }
```

Syntax to create object:

$objName= new ClassName;

OR

$objName= new ClassName(arg list);

Q.10. Illustrate class inheritance in PHP with example. (W-22) 6 Marks

Ans.

- Inheritance is a mechanism of extending an existing class to define new derived class.
- Where a newly created or derived class has all functionalities of existing class along with its own properties and methods.
- The parent class is also called a base class or super class. And the child class is also known as a derived class or a subclass.

- Inheritance allows a class to **reuse the code** from another class without duplicating it.
- Reusing existing codes serves various advantages. It **saves time, cost, effort, and increases a program's reliability.**
- To define a class inherits from another class, use the extends keyword.
- **Types of Inheritance:** Single Inheritance Multilevel Inheritance Multiple Inheritance Hierarchical Inheritance
- Example:

```
<?php
class User
{
public $name;
public $gender;
function __construct($n,$g)
{
$this->name= $n;
$this->gender= $g;
}
}
class Employee extends User
{
public $id;
public $salary;
public $designation;
function __construct($n,$g,$i,$s,$dg)
{
$this->name= $n;
$this->gender= $g;
$this->id= $i;
$this->salary= $s;
$this->designation= $dg;
}
function getEmployeeDetails()
{
Echo "Name: $this->name<br>Gender:
$this->gender<br>Employee Id:
```

```
$this->id<br>Salary:$this->salary<br>
designation=$this->designation";
}
}
 $e1 = new Employee("Sneha","F",114,34555,"Manager");
$e1->getEmployeeDetails();
?>
```

Q.11 State the use of serialization. (W-22, S-23) 3 Marks, (S-24) 2 Marks

Ans.

Serializing an object means converting it to a byte stream representation that can be stored in a file.

Serialize() :

The serialize() converts a storable representation of a value.

The serialize() function accepts a single parameter which is the data we want to serialize and returns a serialized string.

A serialize data means a sequence of bits so that it can be stored in a file, a memory buffer or transmitted across a network connection link.

It is useful for storing or passing PHP values around without losing their type and structure.

Syntax : serialize(value1);

Example:

```
<?php
$s_data= serialize (array ('Welcome', 'to', 'PHP'));

// printing the serialized data
print_r($s_data . "");
$newvar = unserialize ($string);

// printing the unserialized data
print_r($newvar);
?>
```

OUTPUT:

```
a:3:{i:0;s:7:"Welcome";i:1;s:2:"to";i:2;s:3:"PHP";}
Array ( [0] => Welcome [1] => 568 [2] => Array ( [0] => 1 [1] => two ) [3] => VP )
```

Q.12. State the query to insert data in the database. (W-22) 3 Marks

Ans.

```
<?php
require_once ‘login.php’;
$conn= newmysqli ($hostname,$username,$password, $dbname);
$querry = INSERT INTO studentinfo (rollno,name,percentage) VALUES(114,“Sneha Patange”,                    95.6)”;
            $result= $conn->query($query);
             if (!$result)
            die ("Database access failed: ". $conn->error);
             else
            echo "record inserted successfully";
            ?>
```

Output: record inserted successfully

Q.13. Describe inheritance, overloading, overriding and cloning object. (S-23) 4 Marks

Ans.

Inheritance allows classes to inherit properties and methods from a parent class, overloading provides dynamic property and method handling, overriding enables customization of inherited methods, and cloning creates copies of objects.

1. **Inheritance:** Inheritance is a fundamental concept in object-oriented programming that allows classes to inherit properties and methods from another class. In PHP, we can define a new class by extending an existing class using the extends keyword. The new class is called the child or derived class, and the existing class is called the parent or base class. The child class inherits all the public and protected properties and methods of the parent class. It can also add its own properties and methods or override the parent class's methods.
2. **Overloading:** In PHP, overloading refers to the ability to dynamically create properties and methods in a class at runtime.

There are two types of overloading:

a. **Property Overloading:** PHP provides the __set() and __get() magic methods to handle property overloading. When a property is accessed or modified that doesn't exist or is inaccessible within the class, these methods are called, allowing us to define custom logic for handling the property.

b. **Method Overloading:** PHP doesn't support method overloading in the traditional sense (having multiple methods with the same name but different parameters). However,we can use the __call() magic method to handle method overloading. It gets called when a non-existent or inaccessible method is invoked, giving the flexibility to handle the method dynamically.

3. **Overriding**: Overriding occurs when a child class implements a method already defined in the parent class. The method signature (name and parameters) in the child class must match that of the parent class. By overriding a method, we can customize the behaviors of the method in the child class while retaining the same method name. To override a method in PHP, simply declare the method in the child class with the same name as the parent class's method.
4. **Cloning Objects:** Cloning an object in PHP allows us to create a duplicate of an existing object. The clone is a separate instance of the class, but its properties will initially have the same values as the original object. In PHP, we can clone an object using the clone keyword followed by the object you want to clone. The cloning process involves calling the __clone() magic method if it's defined in the class. This method allows us to customize the cloning process by modifying the properties of the cloned object if necessary.

Q.14 Explain the concept of constructor and destructor in detail. **(S-23) 6 Marks**
Demonstrate __--construct() and __destruct() in PHP **(S-24,W-24) 4 Marks**

Ans.

Constructor:

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 arhument
To define constructor, 'construct' method is used with two underscores (__).

Syntax:

```
function __construct([argument1, argument2, ..., argumentN]) {
/* Class initialization code */
 }
```

Destructor:

A destructor function in class is used to destroy an object created by an constructor.
A destructor function is commonly called in two ways: When a script ends or manually delete an object with the unset() function.The 'destruct' method starts with two underscores (__).

Syntax :

```
function __destruct() {
/* 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");
?>
```

OUTOUT:
Constructor is executed for Sneha
Destructor is executing for Sneha

Explain interface with example W-24

In PHP, interfaces are used to define a contract for classes. An interface specifies a set of methods that a class must implement, but it does not provide any implementation for those methods. This allows you to define a common set of methods that multiple classes can implement, ensuring consistency across different implementations.

Syntax for Defining an Interface

```php
interface InterfaceName {
    public function method1();
    public function method2($param);
    // More methods...
}
```

Implementing an Interface

A class can implement an interface using the `implements` keyword. The class must provide concrete implementations for all the methods defined in the interface.

```php
class ClassName implements InterfaceName {
    public function method1() {
        // Implementation of method1
    }

    public function method2($param) {
        // Implementation of method2
    }
```

}
```

### Example

Let's create an interface `Shape` with a method `calculateArea()`. Then, we'll create two classes `Circle` and `Rectangle` that implement this interface.

```php
<?php

// Define the Shape interface
interface Shape {
    public function calculateArea();
}

// Implement the Shape interface in the Circle class
class Circle implements Shape {
    private $radius;

    public function __construct($radius) {
        $this->radius = $radius;
    }

    public function calculateArea() {
        return pi() * pow($this->radius, 2);
    }
}

// Implement the Shape interface in the Rectangle class
class Rectangle implements Shape {
    private $width;
    private $height;

    public function __construct($width, $height) {
        $this->width = $width;
        $this->height = $height;
    }

    public function calculateArea() {
        return $this->width * $this->height;
    }

```
}

// Create instances of Circle and Rectangle
$circle = new Circle(5);
$rectangle = new Rectangle(4, 6);

// Calculate and display the areas
echo "Area of Circle: " . $circle->calculateArea() . "\n";
echo "Area of Rectangle: " . $rectangle->calculateArea() . "\n";

?>
```

Output

```
Area of Circle: 78.539816339745
Area of Rectangle: 24
```

UNIT 4 Creating and validating Forms

Q.1 How can we destroy cookies. (S-22) 2 Marks

Ans:

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

Syntax:

setcookie(name, time() - 3600);

Example :

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

Q.2 How to create session variable in PHP? (S-22) 2 Marks

Ans:

- Session variable can be get with a help of a PHP global variable: $_SESSION.
- While accessing the data using $_SESSION variable we have to mention key in the $_SESSION variable.
- For example, we want retrieve the data stored in session variable.

Example 1 :

```
<?php
session_start();
echo “User name : ”$_SESSION["username"];
?>
```

Q.3 Write difference between get( ) & post( ) method of form (Any four points). (S-22) 4 Marks
Ans:

Parameters	GET	POST
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-url encoded	application/x-www-form-url encoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed

Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

Q.4 Define session & cookie. Explain use of session start. (S-22) 4 Marks

Ans :

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

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

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.

Q.5 Create customer form like customer name, address, mobile no, date of birth using different form of input elements & display user inserted values in new PHP form. (S-22) 4 Marks

Ans:

```
<head>
<title>Registration form</title>
<link rel="stylesheet" href="styel.css" type="text/css">
</head>
<body>
<div class="main">
<div class="register">
<h2>Register Here</h2>
<form id="register" action="registrationform.php" method="post">
<label>First name:</label>
<br>
```

```
<input type="text" id="fname" name="fname" placeholder="Enter your first name">
<br>
<br>
<label>Last name:</label>
<br>
<input type="text" id="lname" name="lname" placeholder="Enter your last name">
<br>
<br>

<label for="email">Enter your email:</label>
<br>
<input type="email" id="email" name="email" placeholder="Enter valid email">
<br>
<br>
<label>Date of Birth:</label>
<br>
<input type="date" id="dob" name="dob" placeholder="Enter your Birthday">
<br>
<br>
<label>Suggestion:</label>
<textarea name="data" id="data" cols=”50” rows=”5”>
</textarea>
<label>Enter mobile number:</label><br><br>
<input type="tel" id="mobno" name="mobno" placeholder="Enter your mobile number" pattern="[7-9]{1}[0-9]{9}" required>
<br>
<br>
<label>Select Usertype</label>
<br>
<select name="usertype" id="usertype">
<option value="Student">Student</option>
```

```
<option value="Staff">Staff</option>
<option value="Alumni">Alumni</option>
</select>
<br>
```

Create a web page using GUI components

```
<!DOCTYPE html>
<html>
<head>
<title>Registration form</title>
<link rel="stylesheet" href="styel.css" type="text/css">
</head>
<body>
<div class="main">
<div class="register">
<h2>Register Here</h2>
<form id="register" action="registrationform.php" method="post">

<label>First name:</label>
<br>
<input type="text" id="fname" name="fname" placeholder="Enter your first name">
<br>
<br>
<label>Last name:</label>
<br>
<input type="text" id="lname" name="lname" placeholder="Enter your last name">
<br>
<br>
<label for="email">Enter your email:</label>
<br>
<input type="email" id="email" name="email" placeholder="Enter valid email">
```

```
<br>
<br>
<label>Date of Birth:</label>
<br>
<input type="date" id="dob" name="dob" placeholder="Enter your Birthday">
<br>
<br>
<label>Enter mobile number:</label><br><br>
<input type="tel" id="mobno" name="mobno" placeholder="Enter your mobile number" pattern="[7-9]{1}[0-9]{9}" required>
<br>
<input type="submit" id="submit" name="submit" value="Submit">
</form>
</div><!-----end of register--->
</div><!-----end of main --->
</body>
</html>
```

Q.6 How do you validate user inputs in PHP ? (S-22) 4 Marks

Ans : A web page having multiple forms

- 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" /><br/>
Mobile Number :<input type="text" name="mobileno" id="mobileno" /><br/>
Email ID :<input type="text" name="email" id="email" /><br/>
<input type="submit" name="submit_btn" value="Submit" />
</form>
</body>
</html>
<?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/>";
}
}
?>
```

Q.7 Write a PHP program to demonstrate use of cookies. (S-22) 6 Marks

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!<br>";
   echo "Value is: " . $_COOKIE[$cookie_name];
}
if(!isset($_COOKIE["user"]))
 {
    echo "Sorry, cookie is not found!";
   } else {
      echo "<br/>Cookie Value: " . $_COOKIE["user"];
   }
setcookie("user"," ",time()-3600);
 echo "Cookie 'user' is deleted.";
```

```
?>
</body>
</html>
```

Q.8 State the use of Cookies.(W-22) 2 Marks

Ans :use of cookies -

- PHP cookie is a small piece of information which is stored at client browser. It is used to recognize the user.
- Cookie is created at server side and saved to client browser. Each time when client sends request to the server, cookie is embedded with request. Such way, cookie can be received at the server side. In short, cookie can be created, sent and received at server end.
- A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.
- The name of the cookie is automatically assigned to a variable of the same name. For example, if a cookie was sent with the name "user", a variable is automatically created called $user, containing the cookie value.

Q.9 State role of GET and POST methods. (W-22) 2 Marks

Ans:

- Get method:It processes the client request which is sent by the client, using the HTTP get method.Browser uses get method to send request.
- Post method -It Handles request in servlet which is sent by the client. If a client is entering registration data in an html form, the data can be sent using post method.

Q.10 Describe : (i) Start session (ii) Get session variable. (W-22) 4 Marks

Ans :

- 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

Syntax 1-
Bool session_start(void)
Example 1.session_start();
PHP $_SESSION is an associative array that contains all session
variables. It is used to set and get session variable values.

Example: Store information
2. $_SESSION["CLASS"] = "TYIF STUDENTS“
Example: Program to set the session variable (demo_session1.php)

```
<?php
session_start();
?>
<html>
<body>
<?php
$_SESSION["CLASS"] = "TYIF STUDDENTS";
echo "Session information are set successfully.<br/>";
?>
</body>
</html>
```

ii)Get Session variables-

- We create another page called "demo_session2.php".
- we will access the session information we set on the first page ("demo_session1.php").
- notice that session variables are not passed individually to each new
- page, instead they are retrieved from the session we open at the
- beginning of each page (session_start()).
- Also notice that all session variable values are stored in the global
- $_SESSION variable:
- Example:- program to get the session variable
- values(demo_session2.php)

```
<?php
session_start();
?>
<html>
<body>
<?php
echo "CLASS is: ".$_SESSION["CLASS"];
?>
```

</body> </html>

Q.11 State any four form controls to get user's input in PHP. (W-22) 4 Marks

Ans :following are from controls to take input from user -

1. Textbox control:It is used to enter data. It is a single line input on a web page.
Tag :<input type="text">
2. Password control:It is used to enter data that appears in the form of special characters on a web page inside box. Password box looks like a text box on a wab page.
Tag:<input type="password">
3. Textarea : It is used to display a textbox that allow user to enter multiple lines of text.
Tag :<textarea> … </textarea>
4. Checkbox:It is used to display multiple options from which user can select one or more options.
Tag: <input type="checkbox">
5. Radio / option button :These are used to display multiple options from which user can select only one option.
Tag :<input type="radio">
6. Select element (list) / Combo box / list box:
<select> … </select> : This tag is used to create a drop-down list box or scrolling list box from which user can select one or more Options.
<option> … </option> tag is used to insert item in a list.

Q.12 Write steps to create webpage using GUI components. (W-22) 4 Marks

Ans : Following are the GUI components to design web page:
Button - has a textual label and is designed to invoke an action when pushed.
 Checkbox - has textual label that can be toggled on and off.
 Option - is a component that provides a pop-up menu of choices.
 Label - is a component that displays a single line of read-only, non-selectable text.

 Scrollbar - is a slider to denote a position or a value.
 TextField - is a component that implements a single line of text.
 TextArea - is a component that implements multiple lines of text.

To design web pages in PHP:
Step 1) start with <html>
Step 2) If user required to add CSS in <head> section.

```
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
```

Step 3) In <body> section design form with all mentioned components.
Step 4) using <?php
Write script for validation for all required input field.
Save the file with php extension to htdocs (C:/Program Files/XAMPP/htdocs)
Note: You can also create any folders inside 'htdocs' folder and save our codes over there.
Step 5) Using XAMPP server, start the service 'Apache'.
Step 6)Now to run your code, open localhost/abc.php on any web browser then it gets executed.

Q.13 Write a PHP program to set and modify cookies. (W-22) 6 Marks

Ans - PHP program to set cookies

```
<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!<br>";
echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
</body>
</html>
```

Output:
Cookie 'username' is set!
Value is: abc
PHP program to modify cookies

```
<?php
setcookie("user", "xyz");
?>
<html>
<body>
<?php
if(!isset($_COOKIE["user"]))
{
echo "Sorry, cookie is not found!";
} else {
echo "<br/>Cookie Value: " .
$_COOKIE["user"];
}
?>
</body>
</html>
```

Output:
Cookie Value: xyz

Q.14 Enlist the attributes of cookies.(S-23) 2 Marks
Ans: Attributes of Cookies are as follows:

1. name

2. value
3. expire
4. path
5. Domain
6. Secure

Q.15 Define GET & POST methods. (S-23) 2 Marks

Ans : GET method:It processes the client request which is sent by the client, using the HTTP get method. Browser uses get method to send request.

– POST method- It Handles request in servlet which is sent by the client. If a client is entering registration data in an html form, the data can be sent using post method.

Q.16 Describe the procedure of sending email. (S-23) 4 Marks

Ans :

– PHP uses Simple Mail Transmission Protocol (SMTP) to send mail.
– Settings of the SMTP mail can be done through “php.ini” file present in the PHP installation folder.
– Any text editor will be used to open and edit “php.ini”file.
– Locate [mail function] in the file.
– After [mail function]in the file following things will be displayed :
o Don’t remove the semi column if you want to work with an SMTP Server like Mercury
o ; SMTP = localhost;
o ; smtp_port = 25
– Remove the semi colons before SMTP and smtp_port and set the SMTP to your smtp server and the port to your smtp port. Your settings should look as follows :
o SMTP = smtp.example.com
o smtp_port = 25
o We can get your SMTP settings from your web hosting providers.
o If the server requires authentication, then add the following lines :

1. auth_username = example_username@example.com
2. auth_password = example_password
3. Save the new changes.

4. Restart server.

Parameters	Descriptions
to	Required. Specifies the receiver or receivers email ids.
subject	Required. Specifies the subject of the email. This parameter cannot contain any newline characters.
message	Required. Defines the message to be sent. Each line should be separated with a (\n). Lines should not exceed 70 characters.
headers	Optional. Specifies additional headers, like From, Cc, and Bcc.
parameters	Optional. Additional parameter to the send mail can be mentioned in this section.

PHP Mail

```
<html>
<head>
<title>Email using PHP</title>
</head>
<body>
<?php
    $to = "xyz@abcdomain.com";
    $subject = "This is subject";
    $message = "<b>This is HTML message.</b>";
    $header = "From:abc@xyzdomain.com \r\n";
    $header .= "Cc:pqr@xyzdomain.com \r\n";
    $header .= "MIME-Version: 1.0\r\n";
    $header .= "Content-type: text/html\r\n";
    $retvalue = mail ($to,$subject,$message,$header);
```

```
if( $retvalue == true ) {
        echo "Message sent successfully...";
}else {
        echo "Message could not be sent...";
      }
   ?>
</body>
</html>
```

Q.18 Describe the procedure of validation of web page. (S-23) 4 Marks

Ans :

- Validating a web page in PHP involves checking the input data provided by users to ensure that it meets the required criteria and is safe for further processing. Here's a general procedure for validating a web page in PHP:
- **The procedure of validation of web page.**
  - **Define the Validation Rules**: Determine the validation rules for each input field on the web page. This includes constraints such as required fields, data formats (e.g., email, date), length limits, and any specific patterns or restrictions.
  - **Create the HTML Form**: Design and create the HTML form that collects user input. Specify the appropriate input types, such as text, email, number, etc., and include any necessary attributes like required or pattern.Submitting the Form: Set up the PHP script that processes the form submission. This script will be responsible for handling the validation and processing the data. Ensure that the form's method attribute is set to "POST" so that the data is sent securely.
  - **Retrieve and Sanitize Input**: In the PHP script, retrieve the submitted data using the $_POST super global array. Sanitize the input to remove any unwanted characters or tags that could potentially pose security risks. You can use functions like html special chars or filter_input to sanitize specific inputs.
  - **Perform Validation:** Apply the validation rules defined in Step 1 to each input field. Use PHP's conditional statements, loops, and regular expressions to check if the input data meets the required criteria. For example, you can use if statements and regular expressions to validate email addresses, check for empty fields, or validate numeric values.
  - **Display Validation Errors**: If any input fails validation, store the error messages in an array or variable. Display these error messages next to the corresponding input fields on the web page, informing the user about the specific validation issues they need to address.
  - **Process Valid Data:** If all the input data passes validation, proceed with further processing, such as storing the data in a database, sending emails, or performing other necessary operations.
  - **Redisplay Form with Pre-filled Data:** If there are validation errors, redisplay the form with the user's previously submitted data already filled in. This provides a better user experience and allows users to correct the invalid fields without re-entering all the data.

Example:

```
if($_SERVER["REQUEST_METHOD"] == "POST"){
   // Validate name
```

```
    if(empty(trim($_POST["name"]))){
        $name_err = "Please enter your name.";
    } else{
        $name = trim($_POST["name"]);
    }
}
```

Q.19 Explain web server role in web development. (S-23) 4 Marks

Ans :

– A client (browser) submits an HTTP request using GET and POST method to the server.

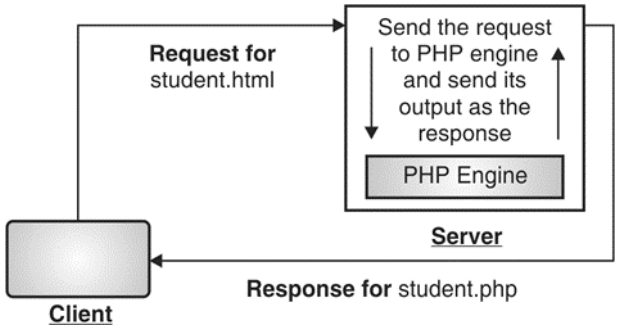
– Server processes the request and returns a result, in the form of a response to the client.

– The response from the server contains status information about the request and requested content.

– Using PHP you can create dynamic web sites.

– Here student.html has a static behavior because it delivers same content always. But student.php has a dynamic behavior because the content it delivers changes according to the time of the day. You can improve this file and have different greetings for different time periods.

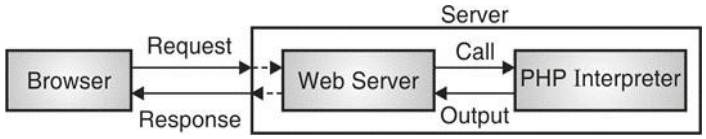
– When PHP interpreter reads a file, it only processes lines between <?php and ?> tags. It outputs rest of the lines without any processing.



PHP Script Life cycle consists of following steps :

1. Client Request
2. Server Parse Request and Send Response

3. Client interpret Responded Text/HTML etc…



– We always start with a browser making a request for a web page. This request is going to hit the web server. The web server will then analyze it and determine what to do with it.

– If the web server determines that the request is for a PHP file (often index.php), it will pass that file to the PHP interpreter. The PHP interpreter will read the PHP file, parse it (and other included files) and then execute it. Once the PHP interpreter finishes executing the PHP file, it will return an output. The web server will take that output and send it back as a response to the browser.

Q.20 Create a web page using GUI component. (S-23) 4 Marks

Ans : Create a web page using GUI components

```
<!DOCTYPE html>
<html>
<head>
<title>Registration form</title>
<link rel="stylesheet" href="styel.css" type="text/css">
</head>
<body>
<div class="main">
<div class="register">
<h2>Register Here</h2>
<form id="register" action="registrationform.php" method="post">
<label>First name:</label>
<br>
<input type="text" id="fname" name="fname" placeholder="Enter your first
name">
<br>
<br>
<label>Last name:</label>
```

```
<br>
<input type="text" id="lname" name="lname" placeholder="Enter your last
name">
<br>
<br>
<label for="email">Enter your email:</label>
<br>
<input type="email" id="email" name="email" placeholder="Enter valid
email">
<br>
<br>
<label>Date of Birth:</label>
<br>
<input type="date" id="dob" name="dob" placeholder="Enter your
Birthday">
<br>
<br>
<label>Gender:</label>
<br>
   
<input type="radio" id="male" name="gender" value="male">
 
<body>
<div class="main">
<div class="register">
<h2>Register Here</h2>
<form id="register" action="registrationform.php" method="post">
<label>First name:</label>
<br>
<input type="text" id="fname" name="fname" placeholder="Enter your first
name">
<br>
<br>
<label>Last name:</label>
```

```
<br>
<input type="text" id="lname" name="lname" placeholder="Enter your last
name">
<br>
<br>
<label for="email">Enter your email:</label>
<br>
<input type="email" id="email" name="email" placeholder="Enter valid
email">
<br>
<br>
<label>Date of Birth:</label>
<br>
<input type="date" id="dob" name="dob" placeholder="Enter your
Birthday">
<br>
<br>
<label>Gender:</label>
<br>
   
<input type="radio" id="male" name="gender" value="male">
 
</div><!-----end of main --->
</body>
</html>
```

Q.21 Design form controls –text box, text area, radio button, check box, list and buttons (S-23) 6 Marks

Ans :

Form Controls :

1. Textbox : A text box is used to enter data. It is a single line input on a web page.

Tag :<input type=“text”> : It is used to display a text box on a web page.

Attributes of <input> tag used with text box:
name=“ text” : Specify name of text box for unique identification.
maxlength=number : Specify maximum number of characters that can be accepted in a textbox.
size=number : Specify the width of text box in number of characters.
value=“text” : Specify default text value that appears in the text box when loaded on a web page.
Example:<input type=“text” name=“n1” maxlength=20 size=15 value=“Enter your name” >
2. Textarea : It is used to display a textbox that allow user to enter multiple lines of text. Scrollbar is used to move up and down as well as left and right if the contents are more than size of box.
Tag :<textarea> … </textarea> : It is used to display a multiline text box on a web page.
Attributes:
name=“ text” : Specify name of the element for unique identification.
cols=number : Specify width of the text area.
rows=number : Specify height of the text area.
readonly : Specify a text area as read only element.
Example:<textarea name=“t1” cols=10 rows=10>Enter your suggestions</textarea>
3. Radio / option button : Radio buttons are used to display multiple options from which user can select only one option. When a radio button is selected by user, a dot symbol appears inside button. Multiple option buttons are group together to allow user to select only one option from the group. A group can be created by giving same name to all option buttons in that group.
Tag :<input type=“radio”> : It is used to display a radio button on a web page.
Attributes of <input> tag used with radio button:
name=“ text” : Specify name of radio button for unique identification.
value=“text” : Specify value to be returned to the destination if that radio button is selected.
checked: Specify default selection
Example:<input type=“radio” name=“r1” value=“male”>male

<input type="radio" name="r1" value="female" checked>female

4. Checkbox : Checkbox elements are used to display multiple options from which user can select one or more options. When a checkbox is selected by user, a tickmark(√) symbol appears inside box.
Tag: <input type="checkbox"> : It is used to display a checkbox on a web page.
Attributes of <input> tag used with checkbox:
name=" text" : Specify name of checkbox for unique identification.
value="text" : Specify value to be returned to the destination if that checkbox is checked.
Checked: Specify default selection.
Example: <input type="checkbox" name="c1" value="pen" checked>pen
5. Select element (list) :<select> … </select> : This tag is used to create a drop-down list box or scrolling list box from which user can select one or more options.
Attributes:
name=" text" : Specify name of the element for unique identification.
size=number : Specify number of options visible in a list box on a web page.
Multiple : Allow user to select multiple option with control key.
<option> … </option> tag is used to insert item in a list.
Attributes:
value="text" : Specify value to be sent to the server once selected by user.
selected: Specify default selection.
Example:<select name="s1" size=2>
<option value=" Pizza">Pizza1 </option>
<option value=" Burger">Burger</option>
<option value=" Chocolate">Chocolate</option>
</select>
6. Note: Explanation of button OR submit button OR reset button shall be considered
i) Button : Buttons are used to display a command button which user can click on web page to perform some action.
Tag :<input type="button"> : It is used to display a button on a web page.

Attributes of <input> tag used with button:
name=“ text” : Specify name of button for unique identification.
value=“text” : Specify value to be displayed on button.
Example:<input type=“button” name=“b1” value=“login”>
ii) Submit button : Submit button is used to display a command button which user can click on web page to submit information entered in a form.
Tag :<input type=“submit”> : It is used to display a submit button on a web page.
Attributes of <input> tag used with submit button:
name=“ text” : Specify name of submit button for unique identification.
value=“text” : Specify value to be displayed on submit button.
Example:<input type=“submit” name=“s1” value=“Submit Form”>
iii) Reset button : Reset button is used to clear all elements with their original state after user clicks on it.
Tag :<input type=“reset”> : It is used to display a reset button on a web page.
Attributes of <input> tag used with reset button:
name=“ text” : Specify name of reset button for unique identification
value=“text” : Specify value to be displayed on reset button.
Example: <input type=“reset” name=“r1” value=“Reset Form”>

Q.22 List attributes of cookie.(W-23) 2 Marks
Ans : Attributes of Cookies are as follows:

- name
- value
- expire
- path
- domain

Q.23 Develop a simple procedure of sending email. (S-23) 4 Marks
Ans:

- PHP uses Simple Mail Transmission Protocol (SMTP) to send mail.

- Settings of the SMTP mail can be done through “php.ini” file present in the PHP installation folder.
- Any text editor will be used to open and edit “php.ini”file.
- Locate [mail function] in the file.
- After [mail function]in the file following things will be displayed :
  - Don’t remove the semi column if you want to work with an SMTP Server like Mercury
  - ; SMTP = localhost;
  - ; smtp_port = 25
  - Remove the semi colons before SMTP and smtp_port and set the SMTP to your smtp server and the port to your smtp port. Your settings should look as follows :
  - SMTP = smtp.example.com
  - smtp_port = 25
  - We can get your SMTP settings from your web hosting providers.
  - If the server requires authentication, then add the following lines :

1. auth_username = example_username@example.com
2. auth_password = example_password
3. Save the new changes.
4. Restart server.

Parameters	Descriptions
to	Required. Specifies the receiver or receivers email ids.
subject	Required. Specifies the subject of the email. This parameter cannot contain any newline characters.
message	Required. Defines the message to be sent. Each line should be separated with a (\n). Lines should not exceed 70 characters.
headers	Optional. Specifies additional headers, like From, Cc, and Bcc.
parameters	Optional. Additional parameter to the send mail can be mentioned in this section.

PHP Mail

```
<html>
<head>
<title>Email using PHP</title>
</head>
<body>
<?php
        $to = "xyz@abcdomain.com";
        $subject = "This is subject";
        $message = "<b>This is HTML message.</b>";
        $header = "From:abc@xyzdomain.com \r\n";
        $header .= "Cc:pqr@xyzdomain.com \r\n";
        $header .= "MIME-Version: 1.0\r\n";
        $header .= "Content-type: text/html\r\n";
        $retvalue = mail ($to,$subject,$message,$header);
if( $retvalue == true ) {
           echo "Message sent successfully...";
}else {
           echo "Message could not be sent...";
        }
     ?>
</body>
</html>
```

Q.24 Differentiate between session and cookies (S-23) 4 Marks
Ans:

Cookie	Session
Cookies are stored in browser as text file format.	Sessions are stored in server side.

The cookie is a client-side resource.	The session is a server-side resource.
It is stored limit amount of data.	It is stored unlimited amount of data
It is only allowing 4kb[4096bytes].	It is holding the multiple variable in sessions.
It is not holding the multiple variable in cookies.	It is holding the multiple variable in sessions.
We can accessing the cookies values in easily. So it is less secure.	We cannot accessing the session values in easily. So it is more secure.
Remember info until deleted by user or expiry.	Remembers info until web site time-out.
Setting the cookie time to expire the cookie.	using session_destory(), we will destroyed the sessions.
The setcookie() function must appear BEFORE the <html> tag.	The session_start() function must be the very first thing in your document. Before any HTML tags.
Usually contains an id string.	Usually contains more complex information.
Specific identifier links to server.	Specific identifier links to user.

Q.25 Design web page using following form controls: (i) Radio Button (ii) Check Box (S-23) 4 Marks
Ans:

radiobtndemo.html

```
<html>
<head>
<title>Radio Button Demo</title>
</head>
<body>
<form method="get" action="phpradiobtndemo.php">
<label>Select your Gender:</label><br/>
<input type="radio" name="gender" value="male" checked> Male<br/>
<input type="radio" name="gender" value="female"> Female<br/>
<input type="submit" value="Submit">
</form>
```

```
</body>
</html>
```

phpradiobtndemo.php

```
<?php
if(isset($_GET["gender"])){
   echo "<p>Gender : " . $_GET["gender"] . "</p>";
}
?>
```

Q.26 Write a PHP program to demonstrate session management. (S-23) 6 Marks

Ans :

```
<?php
session_start();
$_SESSION["Rollnumber"] = "11";
$_SESSION["Name"] = "Ajay";
echo 'The Name of the student is :' . $_SESSION["Name"] . '<br>';
echo 'The Roll number of the student is :' . $_SESSION["Rollnumber"] . '<br>';
if(isset($_SESSION["Name"])){
   unset($_SESSION["Rollnumber"]);
}
session_destroy();
?>
```

The session IDs are randomly generated by the PHP engine .
The session data is stored on the server therefore it doesn't have to be sent with every browser request.
The session_start() function needs to be called at the beginning of the page, before any output is generated by the script in the browser.

Unit 5 :Database Operations

Q.1. List any four data types in MySQL. (S – 22) 2 Marks
Ans:

- Numeric data types such as: INT , TINYINT , BIGINT , FLOAT , REAL , etc.
- Date and Time data types such as: DATE , TIME , DATETIME , etc.
- Character and String data types such as: CHAR , VARCHAR , TEXT , etc.
- Unicode character string data types such as: NCHAR , NVARCHAR , NTEXT , etc

Q.2. Explain delete operation of PHP on table data. (S – 22) 4 Marks
Ans:

- 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();
?>
```

Q.3. Inserting and retrieving the query result operations. (S – 22) 4 Marks

Ans:

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

Q.4 How do you connect MySQL database with PHP. (S – 22) 6 Marks

Ans:

Connecting MySQL Database Server from PHP

- The Process : The process of using MySQL with PHP is as follows :

1. Connect to MySQL and select the database to use.
2. Build a query string.
3. Perform the query.
4. Retrieve the results and output them to a web page.
5. Repeat steps 2 to 4 until all desired data has been retrieved.
6. Disconnect from MySQL.

Creating a Login File

- When a web site is developed with PHP it contains multiple program files that will require access to MySQL and will thus need the login and password details.
- Therefore, it is good to create a single file to store login credentials and then include that file wherever it is needed.

Example . The login.php file

```
<?php
$hn = 'localhost';
$db = 'college';
```

```
$un = 'root';
$pw = '';
?>
<?php
require_once 'login.php';
$conn = new mysqli($hn, $un, $pw, $db);
if ($conn->connect_error) die($conn->connect_error);
?>
```

Q.5 (ii) Write update operation on table data. (S – 22) 3 Marks

Ans:

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

Q.6. List two database operations. (W -22) 2 Marks

Ans:

i)select
ii)insert

iii)update
iv)delete

Q.7 Write steps to create database using PHP. (W -22) 4 Marks
Ans:
We can create database in php using

- MySQLi (object-oriented)
- MySQLi (procedural)
- PDO(php data object)

A database consists of one or more tables.You will need special CREATE privileges to create or to delete a MySQL database.

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

Q.8 Explain queries to update and delete data in the database. (W -22) 4 Marks
Ans:

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

Q.9 Write a program to connect PHP with MySQL. (W -22) 6 Marks

Ans:

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

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

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

Q.10. State the query to insert data in the database. (W -22) 3 Marks

Ans:

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

Q.11. Write syntax of constructing PHP webpage with MySqL. (S -23) 2 Marks

Ans- Using MySQLi object-oriented procedure:

Syntax:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Creating connection
$conn = new mysqli($servername, $username, $password);
// Checking connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

OR

Using MySQLi procedural procedure :

Syntax:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Creating connection
$conn = mysqli_connect($servername, $username, $password);
// Checking connection
if (!$conn)
```

```
{
die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

Note: Any one relevant syntax shall be considered

Q.12. Write update and delete operations on table data. (S -23,W-22) 4 Marks

Ans:

Update operation

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();
 }
```

```
?>
```

Delete operation-

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

Q.13. Write a program to connect PHP with MySQL. (S -23) (W -23) 4 Marks

Ans:

In this, and in the following chapters we demonstrate three ways of working with PHP and MySQL:

- MySQLi (object-oriented)
- MySQLi (procedural)
- PDO

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

```
$password = "password";

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

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

Q.14 Elaborate the following (S -23) 3 Marks

Ans-

(ii) Mysqli_connect() (S -23)

```
<?php
$mysqli = new mysqli("localhost","my_user","my_password","my_db");
// Check connection
if ($mysqli -> connect_errno)
 {
  echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
  exit();
}
?>
```

Q.15 Define MySQL. (W -23) 2 Marks

Ans:

MySQL is used to manage stored data and is an open source Database Management Software (DBMS) or Relational Database Management System (RDBMS).

Q.16. Explain following Terms (W -23) 4 Marks

(i) POD::_construct()

(ii) mysqli_connect()

Ans:

(i) POD::_construct()

Ans:
Creates a PDO instance to represent a connection to the requested database.

Description

```
public  PDO::_construct(
    string $dsn,
    string $username = null,
    string $password = null,
    array $options = null
)
```

dsn-
The Data Source Name, or DSN, contains the information required to connect to the database.
Username-
The user name for the DSN string. This parameter is optional for some PDO drivers.
Password-
The password for the DSN string. This parameter is optional for some PDO drivers.
Options-
A key=>value array of driver-specific connection options.
(ii) mysqli_connect()
Ans:
Open a new connection to the MySQL server:
Syntax: mysqli_connect(*host, username, password, dbname, port, socket*)
Parameter Values-

host	**Optional. Specifies a host name or an IP address**
username	Optional. Specifies the MySQL username
password	Optional. Specifies the MySQL password
dbname	Optional. Specifies the default database to be used
port	Optional. Specifies the port number to attempt to connect to the MySQL server
socket	Optional. Specifies the socket or named pipe to be used

Q.17 Develop a PHP program to (W -23) 6 Marks

(i) Enter a data into database
(ii) Retrive and present data from database

Ans:

(i) Enter a data into database

```
<?php
require_once 'login.php';
$conn = new mysqli($hn, $un, $pw, $db);
if ($conn->connect_error) die($conn->connect_error);
$query = "INSERT INTO student(rollno,name,percent) VALUES ('CO103','Yogita Khandagale',98.45)";
$result = $conn->query($query);
if (!$result) die ("Database access failed: " . $conn->error);
?>
```

(ii) Retrive and present data from database

```
<?php
require_once 'login.php';
$conn = new mysqli($hn, $un, $pw, $db);
if ($conn->connect_error) die($conn->connect_error);
$query = "SELECT * FROM student";
$result = $conn->query($query);
if (!$result) die ("Database access failed: " . $conn->error);
$rows = $result->num_rows;
echo "<table border='1'><tr><th>Roll No.</th><th>Name</th><th>Percentage</th></tr>";
for ($j = 0 ; $j < $rows ; ++$j)
{
$result->data_seek($j);
$row = $result->fetch_array(MYSQLI_NUM);
echo "<tr>";
for ($k = 0 ; $k < 3 ; ++$k) echo "<td>$row[$k]</td>";
echo "</tr>";
}
echo "</table>";
?>
```

Q.18 List any two database operations(W-22 2 Marks,S-24 2 Marks)
Ans- select,insert ,update and delete operations

Q.19 Write PHP program to demonstrate database operations using PHP and MySQL. (S-24 6 Marks)

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.

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

Q.20. State the query to insert data in the database. (W-22) 3 Marks

Ans.

```
<?php
require_once 'login.php';
$conn= newmysqli ($hostname,$username,$password, $dbname);
$querry = INSERT INTO studentinfo (rollno,name,percentage) VALUES(114,"Sneha Patange",                    95.6)";
          $result= $conn->query($query);
           if (!$result)
          die ("Database access failed: ". $conn->error);
           else
          echo "record inserted successfully";
          ?>
```

Output: record inserted successfully

Q.21. Create a html form 'result.html' to accept rollno of student using submit button. Write 'result.php' code to check the result of student 'pass' or 'fail'. Create a table result_table in MySQL database 'my-db' with two columns rollno and status also write php code to delete a record result_table.

```
<!DOCTYPE html>
<html>
<head>
   <title>Student Result</title>
</head>
<body>
   <h2>Enter Roll Number</h2>
   <form action="result.php" method="post">
      <label for="rollno">Roll Number:</label>
      <input type="text" id="rollno" name="rollno" required>
      <br><br>
      <button type="submit">Check Result</button>
   </form>
   <br>
```

```
    <a href="delete.php">Delete a Record</a>
</body>
</html>
```

2. result.php (PHP Code)

Code

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $rollno = $_POST["rollno"];

    // Database connection details (replace with your credentials)
    $host = "localhost";
    $username = "your_username"; // Replace with your MySQL username
    $password = "your_password"; // Replace with your MySQL password
    $dbname = "my-db"; // Replace with your database name

    // Create connection
    $conn = new mysqli($host, $username, $password, $dbname);

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

    // Prepare and execute SQL query to check result
    $sql = "SELECT status FROM result_table WHERE rollno = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $rollno);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        $status = $row["status"];
```

```
        echo "<h2>Result for Roll Number: " . $rollno . "</h2>";
        echo "<p>Status: " . $status . "</p>";
    } else {
        echo "<h2>Result for Roll Number: " . $rollno . "</h2>";
        echo "<p>No result found for this roll number.</p>";
    }

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

3. delete.php (PHP Code for Deleting Record)

Code

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $rollno = $_POST["rollno_delete"];

    // Database connection details (replace with your credentials)
    $host = "localhost";
    $username = "your_username"; // Replace with your MySQL username
    $password = "your_password"; // Replace with your MySQL password
    $dbname = "my-db"; // Replace with your database name

    // Create connection
    $conn = new mysqli($host, $username, $password, $dbname);

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

    // Prepare and execute SQL query to delete record
    $sql = "DELETE FROM result_table WHERE rollno = ?";
```

```
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $rollno);
    $stmt->execute();

    if ($stmt->affected_rows > 0) {
        echo "Record deleted successfully!";
    } else {
        echo "No record found for deletion.";
    }

    $stmt->close();
    $conn->close();
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Delete Result</title>
</head>
<body>
    <h2>Delete Result</h2>
    <form action="delete.php" method="post">
        <label for="rollno_delete">Roll Number to Delete:</label>
        <input type="text" id="rollno_delete" name="rollno_delete" required>
        <br><br>
        <button type="submit">Delete</button>
    </form>
    <br>
    <a href="result.html">Go Back</a>
</body>
</html>
```