**PHP-QUESTION BANK**

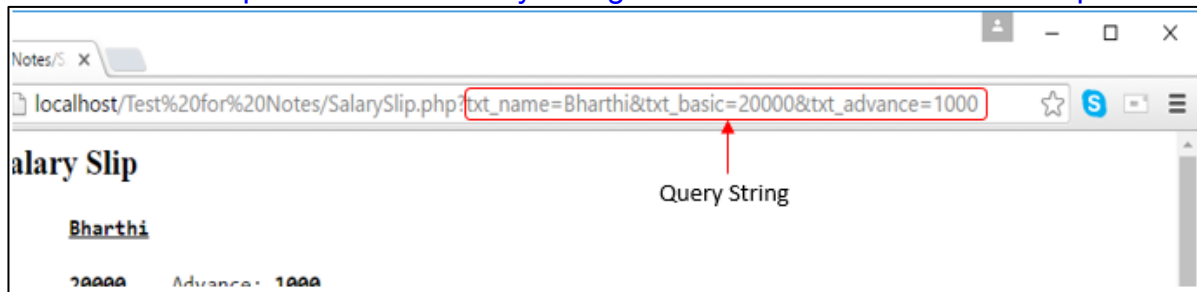| | |
|---|---|
| 1. | **a. Explain the working of QueryString using an example.**<br>When a form is submitted using GET method, the form's data is sent to the server in the form of "variable=value" pairs. A variable and its value are separated by equality sign (=) and different "variable=value" pairs are separated by ampersand sign (&).This complete set of "variable=value" pairs is called Query String and is visible in the URL. Example: |



```
localhost/Test%20for%20Notes/SalarySlip.php?txt_name=Bharthi&txt_basic=20000&txt_advance=1000
```
Query String

alary Slip

**Bharthi**

20000    Advance: 1000

The query string is specified by the values following the question mark (?).

When a form is submitted by GET method, this query string is visible in the URL. A PHP script retrieves this string in $_SERVER["QUERY_SRING"].

Query Strings are used to pass information from a browser to a web server. Several different processes can generate a query string. Query strings are generated by sending a form, by a user typing a query in the address box of the browser or in the anchor tag.

**b. Differentiate between**

**(i)  GET and POST methods of sending data**

| GET | POST |
|---|---|
| Data sent by GET method is restricted to 1024 characters. | In the POST method there is no restriction on data size to be sent. |
| Data is visible in the URL while being sent to the server. Therefore, GET method should not be used to send sensitive data like passwords etc. | Data is invisible as it is embedded within the body of HTTP request. Hence, data being sent is secure. |
| GET cannot be used to send binary data, like images or word documents, to the server. | POST can be used to send ASCII as well as binary data. |

**(ii)  $_GET and $_POST variables.**

$_GET receives the data if a form is submitted by GET method, whereas $_POST receives the data if a form is submitted by POST method.

c. Find the error(s) in the following code:

```php
<?php
$name = $nameErr = "";
OK=false;
if ($_SERVER("REQUEST_METHOD") == "POST")
{
        $name = $_POST["name"];
        if (!empty($name))
                OK=true;
        else
                nameErr = "Please enter name";
}
?>
<html>
<body>
<form name = form1 action=POST
        method = "<?php echo $_SERVER["PHP_SELF"]; ?>" >
Name <input type = textbox name = "name">
<?php echo $nameErr; >
<br>
<input type = submit name = submit value = "SUBMIT">
</form>
<?php
        if $OK==true
                echo "Hello ".$name."!<P>How are you";
?>
</body>
</html>
```

```php
<?php
$name = $nameErr = "";
$OK=false;                      // - 1
if ($_SERVER["REQUEST_METHOD"] == "POST")   // - 2
{
        $name = $_POST["name"];
        if (!empty($name))
                $OK=true;    // - 3
        else
                $nameErr = "Please enter name";       // - 4
}
?>
<html>
<body>
<form name = form1 action method=POST   // - 5
        method action = "<?php echo $_SERVER["PHP_SELF"]; ?>" > // - 6
Name <input type = textbox name = "name">
<?php echo $nameErr; ?>  // - 7
<br>
<input type = submit name = submit value = "SUBMIT">
</form>
<?php
```

```php
                if ($OK==true)        // - 8
                        echo "Hello ".$name."!<P>How are you";
        ?>
        </body>
        </html>
```

**2.** **Create a form which takes the name of the user as input and prints a greeting in the format given below and if the user submits the form with the name field blank, it prints an appropriate error message.**

**Hello Francis!**
**How are you today?**

```php
<?php
$name = $nameErr = "";
$OK=false;
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
        $name = $_POST["name"];
        if (!empty($name))
                $OK=true;
        else
                $nameErr = "Please enter name";
}
?>
<html>
<body>
<form name = form1 method=POST
        action = "<?php echo $_SERVER["PHP_SELF"]; ?>" >
Name <input type = textbox name = "name"> <?php echo $nameErr; ?>
</P>
<input type = submit>
</form>
<?php
        if ($OK)
                echo "Hello ".$name."!</BR>How are you";
?>

</body>
</html>
```

**3.** **Create the following form and based on the user selection print a message in the format given below:**

Please select your favourite car
◉ Nissan
◯ Toyota
◯ Mitsubishi
[ SUBMIT ]

**Your favourite car is: Nissan**

```php
<?php
$car = $carErr = "";
```

```php
    $OK=false;
    if ($_SERVER["REQUEST_METHOD"] == "POST")
    {
        if (!empty($_POST["Car"]))
        {
                $car = $_POST["Car"];
                $OK=true;
        }
        else
                $carErr = "  **Please select a Car";
    }
    ?>
    <html>
    <body>
    <form name = form1 method=POST
        action = "<?php echo $_SERVER["PHP_SELF"]; ?>" >
    Please select your favourite car <?php echo $carErr; ?> <br>
    <input type = radio name = Car value = Nissan>Nissan </BR>
    <input type = radio name = Car value = Toyota>Toyota </BR>
    <input type = radio name = Car value = Mitsubishi>Mitsubishi </BR>
    <input type = submit name = submit value = "SUBMIT">
    </form>
    <?php
        if ($OK==true)          // - 6
                echo "Your favourite car is ".$car;
    ?>

    </body>
    </html>
```

| | |
|---|---|
| 4. | **Create the following form and do form validation to ensure that all the fields are filled/selected by the user. When the user submits the form, the message "Order Placed" with the details of the order must be displayed.** |

## Make Your Pizza

Size of the Pizza
- ○ Small 8" -Serves 1 person
- ○ Medium 12" -Serves 2 persons
- ○ Large 16" -Serves 4 persons

Select Type [Regular ▼]

Toppings: ☐ Cheese ☐ Corn ☐ Tomato ☐ Chicken

## Contact Details

Name: [_____]

Address: [_____]

Phone Number(s): [_____]

[Submit]

```php
<!-- PHP6-FormValidationAns.php -->
<?php
$sizeErr = $typeErr = $topErr = $nameErr = $addressErr = $phoneErr = "";
$size = $type = $name = $address = $phone = "";
$toppings = array();
$OK = false;
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
        $OK=true;
        if (!empty($_POST["size"]))
                $size = $_POST["size"];
        else
        {
                $sizeErr = "Please select pizza Size";
                $OK = false;
        }
        if (!empty($_POST["type"]))
                $type = $_POST["type"];
        else
        {
                $typeErr = "Please select pizza Type";
                $OK = false;
        }
        if (!empty($_POST["Toppings"]))
                $toppings = $_POST["Toppings"];
        else
        {
                $topErr = "Please select 1 or more toppings";
                $OK = false;
        }
        if (!empty($_POST["txtName"]))
```

```php
                    $name = $_POST["txtName"];
            else
            {
                    $nameErr = "Please enter your name";
                    $OK = false;
            }
            if (!empty($_POST["txtAdd"]))
                    $address = $_POST["txtAdd"];
            else
            {
                    $addressErr = "Please enter your Address";
                    $OK = false;
            }
            if (!empty($_POST["txtPhone"]))
                    $phone = $_POST["txtPhone"];
            else
            {
                    $phoneErr = "Please enter your Phone number(s)";
                    $OK = false;
            }
        }
        ?>
        <html>
        <style>
        .error {color:RED}
        </style>
        <body>
        <form name = form1 method = "POST" action =
        "<?php echo $_SERVER['PHP_SELF'] ?>" >
        <h2>Make Your Pizza</h2>
        Size of the Pizza <span class = error><?php echo $sizeErr ?></span> </BR>
        <input type = radio name = size value = Small>Small 8" -Serves 1 person</BR>
        <input type = radio name = size value = Medium>Medium 12" -Serves 2 people</BR>
        <input type = radio name = size value = Large>Large 16" -Serves 4 people
        <P>
        Select Type <?php echo $typeErr ?>
        <Select name = "type">
        <option>Regular</option>
        <option>Thin Crust</option>
        <option>Double Cheese</option>
        </Select>
        <P>
Toppings:
<input type = checkbox name = "Toppings[]" value = Cheese>Cheese
<input type = checkbox name = "Toppings[]" value = Corn>Corn
<input type = checkbox name = "Toppings[]" value = Tomato>Tomato
<input type = checkbox name = "Toppings[]" value = Chicken>Chicken
<BR><?php echo $topErr ?>
<HR>
<h2>Contact Details</h2>
<pre>
```

```php
Name: <input type = textbox name = txtName> <?php echo $nameErr ?><BR>
Address <input type = textbox name = txtAdd><?php echo $addressErr ?><BR>
Phone Number(s): <input type = textbox name = txtPhone> <?php echo $phoneErr
?><BR><input type = submit name = submit>
</form>
</body></html>
<?php
        if (isset($_POST['submit']) && $OK==true)
        {
  echo "<h2>Thanks for placing the order. Your order details are:</h2>";
                echo $size." ".$type." Pizza with ";
                foreach ($toppings as $top)
                        echo $top." ";
                echo "toppings<p>";
                echo "Pizza will be delivered to ".$name." at <br>".$address;
                echo " <br>Phone: ".$phone;
        }
?>
```

| 5. | **What are Superglobals? Give 4 example**<br>**Superglobals are built in Variable in PHP that are always available in all scope and we can access them from any functions class or file**<br>**Main Superglobals are**<br>**$GLOBALS**<br>**$_SERVER**<br>**$_REQUEST**<br>**$_POST**<br>**$_GET**<br>**$_FILES**<br>**$_COOKIE**<br>**$_SESSION**<br>**$_ENV** |
|---|---|
| 6. | **Define preg_match() function ? How to use to check $name contain only letter and space**<br>**preg_match() function searches for pattern, returning true if the pattern exists and false otherwise.**<br>**If(preg_match("/^[a-zA-Z ]*$/",$name))**<br>**echo " valid";**<br>**else   echo "Invalid";** |
| 7. | **Explain how filter_var() function is used to validate Email**<br>**When we use FILTER_VALIDATE_EMAIL filter , in filter_var() function it validate whether the value is a valid email address, returning true if it is valid and false otherwise.**<br>**If(filter_var($email,FILTER_VALIDATE_EMAIL)**<br>**echo " valid";**<br>**else   echo "Invalid";** |
| 8. | **Explain <fome action= "<?php echo $_SERVER["PHP_SELF"]?>" statement**<br>**$_SERVER["PHP_SELF"] is used to  display the path of currently used script file name** |

| | So the above statement action attributes states that after the form submission form data will be handle by the PHP Script present in the same file. |
|---|---|
| 9. | **Explain $_SERVER["REQUEST_METHOD"]**<br>It is used to tell whether data is sent using GET method or POST method when form is submitted |
| 10. | **How can we add PHP in HTML**<br>PHP code can be added to an HTML file by including the code in:<br>☐ the tags <script language=php> and </script><br>☐ the delimiters <?PHP and ?> |
| 11. | **Name two equivalent Tools of PHP**<br>Java Server Page<br>Active Server Page |
| 12. | **Differentiate between** |

| Server Side scripting | Client side scripting |
|---|---|
| Executed by the browser | Executed by the server |
| Actual code can be viewed by the user | Actual code cannot be viewed by the user |
| Creates HTML code | Does not create any code |
| **PHP file** | **HTML file** |
| Has the extension .php | Has the extension .html |
| Contains PHP code and may or may not contain HTML code. | Contains HTML code but does not contain PHP code |
| Cannot be executed by the browser | Can be executed by the browser |
| **Echo** | **Print** |
| Can be used with or without parameters: echoor echo() | Can be used with or without parameters: printor print() |
| Can take multiple parameters separated by comma. | Can take only one parameter. |
| Does not return any value. | Always returns 1 |
| Is faster than print | Is slower than echo. |
| **if..elseif** | **Switch** |
| Multiple conditions involving different expressions may be specified. | Only one expression with different possible values can be specified. |
| Conditions may involve range checking and inequality expressions. | Using switch we can specify only equality conditions comparing one expression with different possible values. |

| While | do..while |
|---|---|
| while specifies an entry- controlled loop | do..while specifies an exit controlled loop. |
| Loop does not execute even once if the condition if false in the beginning. | Loop executes at least once irrespective of the initial value of the condition. |
| | Non-deterministic loop |
| Number of iterations of the loop can be predicted. | Number of iterations of the loop cannot be predicted. |
| The looping condition does not depend on the user | The looping condition depends on the user input. |
| e.g. for loop and foreach loop | While loop and do..while loop. |
| for loop | foreach loop |
| Can be used for arrays or otherwise. | Can be used for arrays only. |
| Need to specify the number of iterations of the loop. | Need not specify the number of iterations of the loop. The loop runs for the number of elements in the specified array. |
| Entry controlled loop | Exit controlled loop |
| Looping condition is specified in the beginning of the loop. | Looping condition is specified at the end of the loop. |
| Loop does not execute even once if the condition if false in the beginning. | Loop executes at least once irrespective of the initial value of the condition. |
| Examples: for loop and while loop | Example: do..while loop |
| date() | getdate() |
| Returns date and time in specified format. | Returns date and time in the form of an associative array. |
| Can take two arguments – date format and timestamp | Can take only one argument – timestamp |
| Local Variable | Global variable |
| A local variable is defined inside a function. | A global variable is defined outside any function. |
| A local variable can be used only in the function in which it is defined. | A global variable can be accessed in any block of the script. |
| Passing by value | Passing by reference |
| It is a copy of corresponding actual parameter | It is an alias of corresponding actual parameter |
| Change in value parameter does not change actual parameter | Change in reference parameter, updates actual parameter |

**13.**

| | |
|---|---|
| **14.** | **Explain the Following Function**<br>a) **strtoupper()** takes a string as argument and returns the string with all alphabetic characters converted to uppercase<br>b) **strtolower**() takes a string as argument and returns the string with all alphabetic characters converted to lowercase.<br>c) **ltrim()** removes whitespaces (or other characters) from the left(beginning) of a given string<br>d) **rtrim()** removes whitespaces (or other characters) from the right(end) of a given string.<br>e) **strcmp()** performs a case-sensitive string comparison whereas strcasecmp() performs a case- insensitive string comparison.<br>f) **sort()** sorts an indexed array in ascending order,<br>g) **rsort()** sorts the array in reverse (descending) order.<br>h) **abs()** function returns the absolute value of specific number . absolute value is always positive. Abs(-4.5) //return 4.5<br>i) **round(number,decimal)** round is used to round the number to no of decimal level. // round(23.43,1) returns 23.4 round(23.44) returns 44<br>j) **pow(x,y)** returns x to the power y pow(2,3) returns $2^3$ =8<br>k) **Sqrt(Number)** returns the square root of number // sqrt(49) returns 7 |
| **15.** | **What is Typecasting how it is done in PHP? Name two function used in PHP for typecasting ?**<br>Converting one data type into other is known as Typecasting. Generally PHP automatically convert one data type into other where ever possible.<br>PHP has two function to do the same also<br>a) **gettype():** it returns a string that represent the name of the data type<br>Syntax   gettype(variable);<br>$ A=17;<br>Print gettype($A);     // returns  integer<br>b) **settype()** it is used to permanently change the data type of variable<br>$Scode=100A<br>settype($Scode, "integer"); //Scode is set to 100 (Integer) |
| **16.** | **What are comments? How to give comments in PHP**<br>Comments are the line ignored by PHP interpreter. Non executable lines<br>Different ways to give comments<br>Using // or # for single line comments<br>/*     */ for Multiple Line Comments |
| **17.** | **What is constant ? Declare a constant Pi and assign value 3.14**<br>Constant is name given to value which will not change during the execution of script. Constant are case sensitive constant are declared using **define** keyword . Constant does no require $ sign to decalre<br>Syntax **define**("Varaiable name",Value);<br>**define**("Pi",3.14); |

| 18. | **Deference between = and ==**<br>**= is assignment operator used to assign value where as == is relational operator used to compare.** |
|---|---|
| 19. | **What is use of ternary(conditional) operator**<br>**Conditional operator(? :) is to used check a condition and produce logical result , Either True or False: The Syntax**<br>**Variable=(condition expression)? Expression for True: Expression for False**<br>**$A =(10%2==0)? "Even No": "Odd No";**<br>**Echo $A;          // Even No** |
| 20. | **Difference between \ and %**<br>**\ is used to find the quotient where as % is used for finding remainder**<br>**\ works with floating value % does not works with Floating** |
| 21. | **What is the use of default and break**<br>**default clause is used to handle the case when no match of any case of switch statement is found.**<br>**break is used to terminate the loop** |
| 22. | **What is an array? How it is differ from ordinary variables?**<br>**An array is collection of variable of same type under one name. Array is defined in the same manner as ordinary variable expect that array variable must accompany by a size**<br>**Array can be created using array( ) function**<br>**Syntax $arrayName=array(Value1,Value2…..ValueN);**<br>**Example  $A=array(2,4,6,8,10);** |

| 23 | **Write the command(s) to open a text file called data.txt which exists in the directory C:\temp.**<br>$file = fopen("c:\temp\data.txt", "r"); |
|---|---|
| 24 | **Explain the PHP fopen() function and PHP fclose() functions.**<br>**fopen() :** fopen() function is used to open a file. The syntax of fopen() is:<br>        fopen(filename, mode);<br>fopen() takes two parameters:<br>  i)      filename: Specifies the name of the file to be opened.<br>  ii)     mode: specifies the mode in which the file should be opened.<br><br>**fclose() :** fclose() is used to close an open file. fclose() takes a single parameter (the file reference) and returns a Boolean value indicating the successful or failed closure of file.<br>Syntax:    fclose(file); |
| 25 | **What are different modes for file opening in PHP?** |

Different file opening modes in PHP are:

| SNo. | Mode | Description |
|------|------|-------------|
| 1. | r | 'Read only' mode. |
| 2. | r+ | Read/Write mode. |
| 3. | w | Write only mode. |
| 4. | w+ | Read/Write mode. |
| 5. | a | Append mode. |
| 6. | a+ | Read/Append mode. |
| 7. | x | Creates the file and opens it in Write only mode. |
| 8. | x+ | Creates the file and opens it in Read/Write mode. |

**26** **Explain with an example how to use PHP fopen() function.**

fopen() function is used to open a file in a specific mode.

**Example:**

$file = fopen("sample.txt", "r");

This command opens a text file names "Sample.txt" in 'Read only' mode.

**27** **How does $_FILES variable work in PHP?**

$_FILES is a superglobal which stores the details of the uploaded files. The details of the each file include

- i)   file name,
- ii)  file type,
- iii) file size (in bytes),
- iv)  temporary location of the file on the server,
- **v)** error code in case some error occurred during of the file upload

**28** **What is the purpose of move_uploaded_file() in PHP?**

When a file is uploaded, it gets stored in a temporary area on the server until it is moved. The file has to be moved from that temporary area, or else it will be destroyed. The function move_uploaded_file() moves an uploaded file to a new location on the server.

**29** **What are PHP cookies? What is the use of a PHP cookie?**

Cookies are small files that the server stores on user's computer when the user visits a website.

Cookies are generally used to identify return visitors, with their approval, so that continuity can be established between visits. Cookies can also be used to keep a user logged into a website indefinitely, track the time of the user's last visit, and much more.

**30** **Explain with an example how to set a PHP cookie.**

A cookie is created with the setcookie() function. The setcookie() function must be run before any other data is sent to the browser, such as the opening <html> tag or random whitespace.

**Example:**

```
<?php
        setcookie("Example", "Anamika", time()+60*60);
?>
```

This code sets a cookie called "Example" that has a value "Anamika". The cookie will expire after one hour (current time + 3600 seconds) of its creation.

**31** **Explain with an example how to delete a PHP cookie?**

To delete a cookie we need to set the same cookie but with no value and with an expiry

date set in the past (i.e., a date that has already expired). This forces the browser to delete the cookie from the user's system.

```php
<?php
//set the expiry date to 1 second ago
setcookie("Example", "", time()-1);
?>
```

| | |
|---|---|
| 32 | **What is the use of setcookie() function in PHP?**<br>The setcookie() function is used to create Cookies. |
| 33 | **What is the importance of sessions used in PHP?**<br>PHP sessions are of small, temporary files that are stored on the website's server. Sessions can be used to store unique information about, or change unique settings for each visitor, but only for the duration of each separate visit. PHP session allows an application to store information for the current session.<br>The information kept in sessions is secure because it is stored on the server, rather than on the client.<br>Session variables store user information to be used across multiple pages (e.g., username, favourite color etc.). By default session variables last until the user closes the browser. |
| 34 | **Explain with an example how to register/start a session in PHP.**<br>A session is started in PHP using Session_start() function. This function must be run on every page, before any other data is sent to the browser, such as the opening <html> tag or random whitespace.<br>Example:<br><?PHP<br>    Session_start();   ?> |
| 35 | **Explain with an example how to destroy a session in PHP.**<br>A session is destroyed by using the session_destroy() function. Session_destry() function ends the session completely, as well as erase all session variables.<br>Example:<br><?php<br>    session_destroy(); ?> |
| 36 | **How is PHP error handling done?**<br>PHP error handling is done in two ways:<br>(i)  Default error handling: The default error handling in PHP is very simple – an error message with filename, line number and a message describing the error is sent to the browser. An example is shown below:<br><br>(!) Notice: Undefined index: uname in C:\wamp\www\PHP5-Example11.php on line 8<br>Call Stack<br>\# Time \| Memory \| Function \| Location<br>1 \| 0.0386 \| 365552 \| {main}( ) \| ..\PHP5-Example11.php:0<br><br>(ii) Custom error handling: To customize the error handling, PHP provides custom error handling techniques and these are:<br>• PHP die() method<br>• Defining custom error handling functions and error triggers<br>• Error reporting or logging |

| 37 | **What are the different PHP error handling methods?**<br>Different PHP error handling methods are:<br>(i)　　Default error handling<br>(ii)　　Custom error handling |
|---|---|

| 38 | **What is PHP die() method?**<br>PHP die() method is used to to overrule the default error handling in PHP. die() method is used to terminate the further execution of PHP script and give a specified error message to the user.<br>Example:<br><br>```php\n<?php\nif (!file_exists("myfile.txt"))\n        die("File not found");\nelse $file = fopen("myfile.txt","r");\n?>\n```<br>This code will check if the file "myfile.txt" exists. If the file does not exist, then it will give a message "File not found" to the user and terminate the further execution of the script. |
|---|---|

| 39 | **Write PHP code to read a text file abc.txt and display alternate lines of the file on the screen.**<br><br>```php\n<!-- PHP7-ReadAlternate.php -->\n<?php\n$file = fopen("abc.txt","r");\nwhile (!feof($file))\n{\n        $line = fgets($file);\n        echo $line."<BR>";\n        $line = fgets($file);\n}\nfclose($file);\n?>\n``` |
|---|---|

| 40 | **Write PHP code to read a text file xyz.txt and count the number of words, lines, and characters in the file.**<br><br>```php\n<!-- PHP7-CountWords.php -->\n<?php\n\n$words = $lines = $char = 0;\n$file = fopen("xyz.txt","r");\nwhile (!feof($file))\n{\n        $data = fgets($file);\n        $lines++;\n}\nfclose($file);\n$file = fopen("xyz.txt","r");\nwhile (!feof($file))\n{\n        $data = fgetc($file);\n        $char++;\n        if ($data == ' ')\n``` |
|---|---|

```php
                $words++;
            if (ord($data) == 13)
            {
                    $words++;
                    $char--;
                    $data = fgetc($file);
            }
}
fclose($file);
$words++;
echo "<P>File contains $words Words, $lines lines, and $char characters";
?>
```

| 41 | **Write PHP code to read a text file str.txt and display all the lines beginning with letter "A".** |
|----|

```php
<!-- PHP7-ShowALines.php -->
<?php
$file = fopen("str.txt","r");
while (!feof($file))
{
        $line = fgets($file);
        if ($line[0] == 'A')
                echo $line."<BR>";
}
fclose($file);
?>
```

| 42 | **Write PHP code to read a text file abc.txt and display the contents of the file on the screen replacing all spaces in the file by the # symbol.** |
|----|

```php
<!-- PHP7-ShowReplaced.php -->
<?php
$file = fopen("abc.txt","r");
while (!feof($file))
{
        $char = fgetc($file);
        if ($char == ' ')
                echo "#";
        else echo $char;
        if (ord($char) == 13)
                echo "<BR>";
}
fclose($file);
?>
```

| 43 | **Write PHP code to read a text file data.txt and count the number of vowels, consonants and other characters present in the file.** |
|----|

```php
<!-- PHP7-CountVowels.php -->
<?php
$file = fopen("data.txt","r");
```

```php
$vowels = $conso = $others = 0;
while (!feof($file))
{
        $char = fgetc($file);
        if (ord($char) != 13 && ord($char) != 10)
        {
                $char = strtoupper($char);
                if ($char=='A' or $char=='E' or $char=='I'
                  or $char=='O' or $char == 'U')
                        $vowels++;
                else if ($char >= 'A' and $char <= 'Z')
                        $conso++;
                else
                        $others++;
        }
}
fclose($file);
echo "File contains $vowels vowels, $conso consonants, and $others other characters";
?>
```

| 44 | **Write PHP code to read a text file named mega.txt and display the contents of the file after converting all the alphabets to upper case alphabets.** |
|---|---|

```php
<!-- PHP7-ConvertToUpper.php -->
<?php
$file = fopen("mega.txt","r");
while (!feof($file))
{
        $char = fgetc($file);
        if (ord($char) == 13)
                echo "<BR>";
        else echo strtoupper($char);
}
fclose($file);

?>
```

| 45 | **What is the difference between PHP fgests() and PHP fgetc()?** |
|---|---|
| | fgets() reads one line of text from the specified file, whereas fgetc() read one character from the specified file. |

| 46 | **How does the PHP execute SQL statements?** |
|---|---|
| | To get the PHP execute the SQL statement using **mysqli_query()** method |

| 47 | **What is use of fetch_assoc() Method** |
|---|---|
| | This method returns an associative array that corresponds to the fetched row, the keys of the element are the name of the columns. |

| 48 | **What is connection object** |
|---|---|

Connection object is used to establish and managing connection between your application and data source

| 49 | **Explain num_row()** |
|----|---|
| | To get the numbers of rows returned by Select Query |

| 50 | **What is the use of Mysql_fetch_array()** |
|----|---|
| | Each time Mysql_fetch_array() is envoked, it returns the next records from the results set as an array. |

---

**51**

## Give the output of the following code :

```php
<?php
$j=5;
if($j>=4)
$j=$j*$j;
else
$j=$j*2;
echo $j
?>
```

```php
<?php
$i=0;
$x=0;
while($i<10)
{
if($i%2 ==0)
{
$x=$x+$i;
echo $x . " ";
}
$i++;
}
?>
```

```php
<?php
$i=0;
for($i=1; $i<=20; $i++)
{
echo $i . " ";
$i=$i+2;
}
?>
```

**Ans 25**                **Ans 0 2 4 6 8**                **Ans 1 4 7 10 13 16 19**

---

**52**

**Change the for Loop to while loop without Affecting output of above**
```php
<?php
$i=1;
$x=0;
for($i=1; $i<10; $i*=2)
{
$x++;
echo $x;
}
echo "<BR>" . $i ;
?>
```

**Ans**
```php
<?php
$i=1;
$x=0;
While($i<10)
{
$x++;
echo $x;
$i*=2
}
echo "<BR>" . $i ;
?>
```

---

**53**

**Change the While Loop to for loop**
```php
<?php
$num=5;
$fact=1;
$i=1;
while($i<=$num) {
$fact=$fact*$i;
echo"<br>" . $fact;
$i++; }
echo "<br><br>FACTORIAL=  " . $fact ;
?>
```

```php
<?php
$num=5;
$fact=1;
$i=1;
For($i=1;$i<=$num;$++) {
$fact=$fact*$i;
echo"<br>" . $fact;
}
echo "<br><br>FACTORIAL=  " . $fact ;
?>
```

| 54 | **Give Output and change the for loop to while loop** | **Give Output and change the while loop to for loop** |
|---|---|---|

**Give Output and change the for loop to while loop**

```
$str = "ABCDE";
for($i=strlen(str); $i>=1;$i--)
{
for($a=0; $a<i; $a++)
{
echo(substr($str,$a));
}
document.write("<br>");
}
```
**//Do it your self**

**Give Output and change the while loop to for loop**

```
var $i=0, $x=0;
while($i<10)
{
if(($i%2)==0)
{
$x=$x+$i
echo($x + " " );
}
$i++;
}// Do it your self
```

| 55 | **Give Output and change the while loop to for loop** | **Give Output and change the while loop to for loop** |
|---|---|---|

**Give Output and change the while loop to for loop**

```
var a, b, c, sum_even=0, sum_odd=0;
$a=20, $b=1;
while($b<=$a)
{ if($b%2==0)
$sum_even+=$b;
else
$sum_odd+=$b;
$b++;
}
echo("sum even No. = " + $sum_even);
echo("sum odd No. = " + $sum_odd);
// Do it your self
```

**Give Output and change the while loop to for loop**

```
$A=1, $B;
while($A<=5)
{ $B=2;
while($B<=$A*$A)
{
echo($B)
$B+=5
} echo("<br>")
$A+=1
}
// Do it your self
```