

COPY OF TEST

SECTION # 1 (9 points)

Note: Attempt any three of the following:

1. Differentiate between server side & client side scripting
2. Explain the term “Markup language” along with CSS with suitable examples
3. What do you know about web 2.0 & HTTP/HTTPS?
4. Differentiate between global & super global variables with examples

SECTION # 2 (7 points)

Note: Fill in the blanks:

1. _____ tag/element is used for sliding text/image in HTML
2. Cookies are stored on _____
3. XML is used for _____
4. Require_once function is used for _____ in php
5. _____ function is mainly used in php to allow downloading of file
6. Destructure function is called when _____
7. Overloading in php can be implemented by _____ function

SECTION # 3 (24 points)

Note: Attempt all of the following:

1. Write a php program that inputs a value from user (via input box) and output result by checking the value that it is prime or not.
2. Write a php program that can communicate & use MySQL database. (assume any dummy db)
3. Differentiate cookies & session by writing their php programs
4. Write a php program that can perform polymorphism

SOLUTION OF TEST

SECTION # 1

1.

Client side scripting

Client-side scripting is changing interface behaviors within a specific web page in response to mouse or keyboard actions, or at specified timing events. In this case, the dynamic behavior occurs within the presentation. The Client-side content is generated on the user's local computer system.

client side coding such as XHTML is executed and stored on a local client (in a web browser) whereas server side code is not available to a client and is executed on a web server which generates the appropriate XHTML which is then sent to the client. The nature of client side coding allows you to alter the HTML on a local client and refresh the pages with updated content (locally).

Server-side scripting

A program running on the web server (server-side scripting) is used to change the web content on various web pages, or to adjust the sequence of or reload of the web pages. Server responses may be determined by such conditions as data in a posted HTML form, parameters in the URL, the type of browser being used, the passage of time, or a database or server state. Such web pages are created with the help of server-side languages.

Dynamic web pages are often cached when there are few or no changes expected and the page is anticipated to receive considerable amount of web traffic that would create slow load times for the server if it had to generate the pages on the fly for each request.

2.

Markup language

A markup language is a modern system for annotating a document in a way that is syntactically distinguishable from the text. The idea and terminology evolved from the "marking up" of manuscripts, i.e., the revision instructions by editors. Markup instructs the software displaying the text to carry out appropriate actions, but is omitted from the version of the text that is displayed to users. Some markup languages, such as HTML, have pre-defined presentation semantics, meaning that their specification prescribes how the structured data are to be presented; a widely used markup language is HyperText Markup Language (HTML), one of the document formats of the World Wide Web.

Cascading Style Sheet

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. It's most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document. CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content.

```

<html>
<head>
<style>
    p { margin-left:20px;}
    body { background-image:url("images/back40.gif");}
</style>
</head>
<body>
    <p> Web Engineering </p>
</body>
</html>

```

3.

Web 2.0

Although Web 2.0 suggests a new version of the World Wide Web, it does not refer to an update to any technical specification, but rather to cumulative changes in the ways software developers and end-users use the Web.

A Web 2.0 site may allow users to interact and collaborate with each other in a social media dialogue as creators of user-generated content in a virtual community, in contrast to websites where people are limited to the passive viewing of content. Examples of Web 2.0 include social networking sites, blogs, wikis, video sharing sites, hosted services, web applications, mashups and folksonomies

HTTP/HTTPS

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems.^[1] HTTP is the foundation of data communication for the World Wide Web.

Hypertext is a multi-linear set of objects, building a network by using logical links (the so-called hyperlinks) between the nodes (e.g. text or words). HTTP is the protocol to exchange or transfer hypertext.

Hypertext Transfer Protocol Secure (HTTPS) is a widely used communications protocol for secure communication over a computer network, with especially wide deployment on the Internet. Technically, it is not a protocol in itself; rather, it is the result of simply layering the Hypertext Transfer Protocol (HTTP) on top of the SSL/TLS protocol, thus adding the security capabilities of SSL/TLS to standard HTTP communications.

4.

Global Variables

In programming paradigm, if any variable is defined inside a function, then it is accessible to only that function (called scope). When any variable is defined on main program (not in any function) then the scope of global variable is valid up to whole program.

Super Global Variables

In PHP, some of pre-defined variables are used which are used to carry data from external contents (or from previous web pages) to current PHP program, such as Cookies, Session, Filing etc.

These Super global variables are denoted by \$_ sign at starting of every super global variable, such as \$_FILE[], \$_GET[] etc.

```

<?php
echo "<form action='even-odd.php' method='post'>
<input type='text' name='insert'>
<input type='submit' name='submit' value='check value'>
</form>";
if(isset($_POST["submit"]))
{
$no = $_POST["insert"];
if($no%2==0)
    echo "No: ".$no." is even";
else
    echo "No: ".$no." is odd";

}
?>

```

SECTION # 2

1. marquee
2. client's computer
3. Sharing data/defining own markup language
4. including external file once
5. header
6. program is terminated/object loses its references
7. __call

SECTION # 3

1. (prime no.php)

```

<?php
echo "<form action='prime no.php' method='post'>
<input type='text' name='insert'>
<input type='submit' name='submit' value='check value'>
</form>";
if(isset($_POST["submit"]))
{
$no = $_POST["insert"];
$flag=0;
if($no%2==0 && $no!=2)
    $flag=1;
else
    for($i=2;$i<=$no/2;$i++)
        if($no%$i==0)
            {

```

```

                $flag=1;
                break;
            }
    if($flag!=0)
        echo "No: ".$no." is not prime";
    else
        echo "No: ".$no." is prime";

}
?>

```

2. (db.php)

```

<?php
$resid=@mysql_connect('localhost','root','') or die("Error in connection: ".mysql_error());
mysql_select_db('we',$resid) or die("Error in selection: ".mysql_error());
$resrec=mysql_query("SELECT * FROM student",$resid)or die("Error in operation: ".mysql_error());
while($row=mysql_fetch_row($resrec))
    echo "Roll No: ".$row[0]." Name: ".$row[1]." Surname: ".$row[2]."<br>";
$no=mysql_num_rows($resrec);
echo "Your query contains ".$no." number of rows";
mysql_close($resid);
?>

```

3. (session.php)

```

<?php
session_start();
if(!isset($_POST["submit"]))
{
    echo "
    <form action='session.php' method='post'>
    Username:<input type='text' name='user'/>
    Password:<input type='password' name='pass'/>
    <input type='submit' name='submit' value='Submit Data'/>
    </form>";
}
if(isset($_POST["submit"]))
{
    $user=$_POST["user"];
    $pass=$_POST["pass"];
    $_SESSION["Username"]=$user;
    $_SESSION["Password"]=$pass;
    if(isset($_SESSION["Username"]))
    {
        echo "Your username is ".$_SESSION["Username"]." & password is ".$_SESSION["Password"];
        session_destroy();
    }
}
?>

```

(cookie.php)

```
<?php
echo "
<form action='cookie.php' method='post'>
Username:<input type='text' name='user'/>
Password:<input type='password' name='pass'/>
<input type='submit' name='submit' value='Submit Data'/>
</form>";
if(isset($_POST["submit"]))
{
$user=$_POST["user"];
$pass=$_POST["pass"];
$i=0;
setcookie("Cookie123",$user."-".$pass."-".$i,0);
if(isset($_COOKIE["Cookie123"]))
{
$arr=explode("-,$_COOKIE["Cookie123"]);
echo "Your username is ".$arr[0]." & password is ".$arr[1];
$arr[2]++;
echo "you have visited ".$arr[2]." times";
setcookie("Cookie123",$arr[0]."-".$arr[1]."-".$arr[2]."-");
}
}
?>
```

4. (overload.php)

```
<?php
class overload
{
function __call($name, $arg)
{
    if($name=="func")
    {
        if (gettype($arg[0])=="string")
            echo "String is: ".$arg[0];

        else if(gettype($arg[0])=="integer")
            echo "Integer value is: ".$arg[0];
    }
}
}
$ov=new overload;
$ov->func("Hello", "World");
$ov->func(25);
?>
```