Final Question Solution

Spring 2018

**1.a)**

<?php require 'connection.php'; ?>

<?php

//for update

if (isset($\_POST['update'])) {

$first\_name = $\_POST['first\_name'];

$age = $\_POST['age'];

$address = $\_POST['address'];

$stmt = $con->query("update registration set first\_name='$first\_name',age='$age',address='$address' where id=1") or die mysqli\_error($con)."at line number ".\_\_LINE\_\_;

if ($stmt) {

echo "Data Updated"}}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Registration Form</title>

<style>

.wrapper{

width: 80%;

margin: 0 auto;

}

</style>

</head>

<body>

<div class="wrapper">

<h1>Registration Form</h1>

<form action="" method="post">

<label for="">First Name</label><br>

<input type="text" name="first\_name"><br>

<label for="">Last Name</label><br>

<input type="text" name="last\_name"><br>

<label for="">Age</label><br>

<input type="number" name="age"><br>

<label for="">Phone</label><br>

<input type="text" name="phone"><br>

<button type="submit">Save</button>

</form>

</body>

</html>

**1.c) Difference between GET and POST**

|  |  |
| --- | --- |
| GET | POST |
| 1. By using GET it is possible to see url parameters | 1. In POST method there no possibility of finding url parameters. |
| 2. GET is used in URL | 2. POST is mainly used for send data from form. |
| 3. Sending parameters by GET method is limited | 3. POST method has not this type of restriction |
|  |  |
|  |  |
|  |  |

**2.a)**

***PHP Session\_\_\_\_***

Sessions are a simple way to store data for individual users against a unique session ID. This can be used to persist state information between page requests. Session IDs are normally sent to the browser via session

***PHP COOKIE\_\_***

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, we can both create and retrieve cookie values.

**2.b)**

**Difference between include and require**

**2.c)**

***Resource:*** A resource is a special variable, holding a reference to an external resource. Resources are created and used by special functions.

***Object:*** Objects. An Object is an individual instance of the data structure defined by a class.

***Static*:** There may be times when a static variable is needed in a PHP function; static variables maintain their value between function calls and are tidier than using a global variable because they cannot be modified outside of the function

***Global*:** Global scope refers to any variable that is defined outside of any function. PHP also stores all global variables in an array called $GLOBALS[index]. Its index is the name of the variable. This array is also accessible from within functions and can be used to update global variables directly.

**3.a)**

**JavaScript Object:** In JavaScript, an object is a standalone entity, with properties and type

**JavaScript Capabilities:**

1. Improving Navigation

2. Validation

3. Special Effects

4. Remote Scripting (Ajax)

**3.b)**

**var a = Number(prompt("Enter first base"));**

**var b = Number(prompt("Enter second base"));**

**var h = Number(prompt("Enter height"));**

**alert(a + b/2) \* h;**

**3.c)**

It is possible to add JavaScript file in html . For this <script> tag is used. Below is an example of adding JavaScript file html

<script src=”lib/main.js”></script>

**4.a**

Object oriented programming main deals with object and class. In class there are several access modifiers such as public, private and protected. Methods and variables of a class are mainly known as property of a class.

public: a method or variable having public keyword can be accessed from anywhere of a project just by making object instance of that class.

private: private method or variable can only be accessed inside of a class.

protected: protected method or variable can only be accessed from child class that represents his parent class.

4.c)

<?php

interface TV\_Programs{

public function setName();

public function setNumber();

}

class Shows implements TV\_Programs

{

private $name;

private $number;

public function \_\_construct($name,$number)

{

$this->setName($name);

$this->setNumber($number);

}

public function setName($name)

{

$this->name = $name;

}

public function setNumber($number)

{

$this->number = $number;

}

public function Air()

{

$data = $this->name.", ".$this->number;

return $data;

}

}

$sh1 = new Shows("Game of Thrones",67);

$sh2 = new Shows("That 70s Show",200);

$sh3 = new Shows("Breaking Bad",62);

$sh1->Air();

$sh2->Air();

$sh3->Air();

?>