Final Question Solution

Fall 2017

1.a)

```
<?php require 'connection.php'; ?>
<?php //for update
function validation($data)
{
	$data = trim($data);
	$data = htmlspecialchars($data);
	$data = stripcslashes($data);
	return $data;
}
$errors = array();
if (isset($_POST['register'])) {

	$first_name = validation($_POST['first_name']);
	$last_name  = validation($_POST['last_name']);
	$age        = validation($_POST['age']);
	if ($first_name == '') {
		array_push($errors, "First name must not be empty<br>");
	} else if(strlen($first_name) < 3) {
		array_push($errors, "First name must be 3 characters
long<br>");
	}
	if ($last_name == '') {
		array_push($errors, "Last name must not be empty<br>");
	}
	if ($age == '') {
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```
            array_push($errors, "Age must not be empty<br>");
      }


      if (count($errors) > 0) {
            foreach ($errors as $error) {
                  echo $error;
      }}}
 ?>
<!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>
                  <button type="submit"
name="register">Register</button>
            </form>
      </div>
</body>
</html>
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

2.b) Exception Handling

During work with any programming languages it is the most responsible duty for a programmer to handle errors or exceptions. Exception handling gives opportunity to show custom error message to the user and loads script. Try and catch are mainly used for handling errors. At first it tries to execute code in try block and if error occurred then a custom error message is thrown. After that, in catch section that custom error message is printed.

Example:

```
try {
        $link = new PDO("mysql:host=localhost;dbname=database",“password”);
        $link->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $link->exec("SET CHARACTER SET utf8");

  } catch (PDOException $exc) {
             die("Field to Connect with Database" . $exc->getMessage())}
```

1.c)

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.

```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30),
"/"); // 86400 = 1 day
?>
```

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

2.a)

```php
<?php
	//demo products 500
	function searchProduct($product)
	{
		$products = array(
			'pro1'=> ['product1',100,'detals1'],
			'pro2'=> ['product2',15,'detals1'],
			'pro3'=> ['product3',150,'detals1'],
			'pro4'=> ['product4',120,'detals1'],
			'pro5'=> ['product5',90,'detals1']
		);
		$status = '';
		foreach ($products as $key =>  $value) {
			foreach ($value as $pro) {
					if($product == $pro){
						$status = "Product Found";
						break;
					}else{
						$status = "Product not found";
					}
			}
		}
		return $status;
	}
	echo searchProduct("product4");
 ?>
```

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

Types of Array

There are three different types of arrays in PHP:

a) **Numeric Array**: An array with a numeric ID key.

b) **Associative Array**: An array where each ID key is associated with a value.

c) Multidimensional Array: An array containing one or more arrays.

Numeric Arrays: Numeric arrays use integer / numbers as their index number to identify each item of the array. The example we discussed above are numeric arrays as they have integer values as index numbers for each item.

```
<?php
$colours = array("white","black","blue");
print_r($colours);
//output will be
      Array
      (
            [0] => white
            [1] => black
            [2] => blue
      )
?>
```

Associative Arrays: Sometimes it's better to use the index name instead of index number for example if you want to save three students' names and numbers so your best option will be to use each student's name as index value for the array and his numbers as the values, behold on the example below

```
<?php
$students['Anna']       = 23;
$students['Maria']      = 24;
$students['Jennifer']   = 25;
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```
/*output will be
Array
      (
            [Anna]      => 23
            [Maria]     => 24
            [Jennifer]  => 25
      )
*/
?>
```

Multidimensional Array: A multidimensional array can contain arrays within itself and the sub arrays contain more arrays within them.

```
<?php
      $david = array(
            'name'       => 'Jhon',
            'address'=> "New York",
            'mobile' => "+12-2654",
            'interest'=> array(
                  'game'          => ['cricket','football','hocky'],
                  'interest'  => ['hiking','travelling']
            )
      );
      echo '<pre>';
      print_r($david);
?>
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

2b).

2c). Loosely Typed Language: PHP is loosely-typed language because in php we can use a variable without declaring it. If a variable is used to store integer then we can still access that integer as string and reverse is possible too. PHP manages that automatically.

Variable Scope

Variable scope is known as its boundary within which it can be visible or accessed from code.

Local variables (local scope)

Global variables (special global scope)

A local scope is a restricted boundary of a variable within which code block it is declared.

```
<?php

function calculate_count() {

        $count = 5;

        //will print 5; the value of local variable

        echo $count++; }

?>
```

As its name, the global scope provides widespread access to the variable declared in this scope.

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```php
<?php

$count = 0;

function calculate_count() {

        global $count;

        // will print 0 and increment global variable

        echo $count++ . "<br/>";

}

calculate_count();

echo $count;

?>
```

3.a) JavaScript and Java are not same at all. JavaScript is a client-side language and runs in browser easily without any kind of extra compiler or interpreter. Whereas Java is a high-level programming language that is not light weight like JavaScript. Java needs extra compiler to turn it into machine readable code. Java is an OOP programming language while JavaScript is an OOP scripting language. Java creates applications that run in a virtual machine or browser while JavaScript code is run on a browser only. Java code needs to be compiled while JavaScript code are all in text. Both They require different plug-ins.

3.b)

```javascript
var a = Number(prompt("Enter height"));
var b = Number(prompt("Enter width"));
alert("Area is "+ a * b);
```

3.c) JavaScript can change different elements in entire html code. It can be changed during page load, after page load or event according to any kind of events (click, mouseover, mouse out etc.). Below is an example……..

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```
<!DOCTYPE html>
<html lang="en">
<head>
	<meta charset="UTF-8">
	<title>Document</title>
</head>
<body>
<span id="show">Old Phrase</span>
<button type="button" onclick="changeText()">Click here</button>
	<script>
		function changeText(){
			document.getElementById("show").innerHTML = 'World';
		}

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


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

4.a) Inheritance is a well-established programming principle, and PHP makes use of this principle in its object model. This principle will affect the way many classes and objects relate to one another

```
<?php
	//inheritance
	class Vehicle{
		private $name;
		private $model;

		public function __construct($name,$model){
			$this->name; = $name;
			$this->model = $model;
		}
		public function printDetails(){
			return "Car name is ".$this->name." and car model is ".$this->model;
		}
	}

	class Car extends Vehicle{
		public function __construct($name,$model){
			parent::__construct($name,$model);
		}
		public function printCar(){
			return "Car name is ".$this->car;
		}
	}
	$v = new Vehicle("Audi","J3jf");
	$c = new Car("Toyota","76xjf");
	$v->printDetails();
	$c->printDetails();
	$c->printCar(); ?>
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

4b). Difference between abstract and interface

OOPs interface vs abstract class

Interface	Abstract class
Interface support multiple implementations.	Abstract class does not support multiple inheritance.
Interface does not contain Data Member	Abstract class contains Data Member
Interface does not contain Constructors	Abstract class contains Constructors
An interface Contains only incomplete member (signature of member)	An abstract class Contains both incomplete (abstract) and complete member
An interface cannot have access modifiers by default everything is assumed as public	An abstract class can contain access modifiers for the subs, functions, properties
Member of interface can not be Static	Only Complete Member of abstract class can be Static

```
<?php
	interface student{
		public function setName($name);
		public function setAge($age);
	}
?>
```

4c).

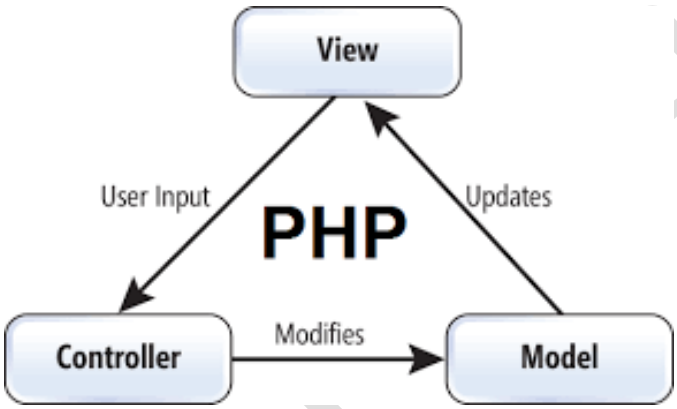
```
<?php
	interface student{
		public function setName($name);
		public function setAge($age);
	}

	class Resource implements student{
```

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```
        private $name;
        private $age;

        public function setName($name)
        {
            $this->name = $name;
        }

        public function setAge($age){
            $this->age = $age;
        }

        public function printData()
        {
            $data  = $this->name;
            $data .= ", ".$this->age;
            return $data;
        }
    }

    $resource = new Resource();
    $resource->setName("Jhon");
    $resource->setAge(20);
    echo $resource->printData();
?>
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

5.a) MVC stands for Model–view–controller. It is a software architectural pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. In MVC pattern when a client requests a server it transfers request to the controller. After that controllers take decision which file will be served for that request. If any database request or query is needed then controller handles it and finally after generating result it shows in browser in html format.



Components of MVC

1) ***Model***: It specifies the logical structure of data in a software application and the high-level class associated with it. It is the domain-specific representation of the data which describes the working of an application. When a model changes its state, domain notifies its associated views, so they can refresh.

2) ***View***: View component is used for all the UI logic of the application and these are the components that display the application's user interface (UI). It renders the model into a form suitable for interaction. Multiple views can exist for a single model for different purposes.

3) ***Controller***: Controllers act as an interface between Model and View components. It processes all the business logic and incoming requests, manipulate data using the Model component, and interact with the Views to

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

render the final output. It receives input and initiates a response by making calls on model objects.

Benefits of MVC

1) Faster development process: MVC supports rapid and parallel development. With MVC, one programmer can work on the view while other can work on the controller to create business logic of the web application. The application developed using MVC can be three times faster than application developed using other development patterns.

2) Ability to provide multiple views: In the MVC Model, you can create multiple views for a model. Code duplication is very limited in MVC because it separates data and business logic from the display.

3) Support for asynchronous technique: MVC also supports asynchronous technique, which helps developers to develop an application that loads very fast.

4) Modification does not affect the entire model: Modification does not affect the entire model because model part does not depend on the views part. Therefore, any changes in the Model will not affect the entire architecture.

5) MVC model returns the data without formatting: MVC pattern returns data without applying any formatting so the same components can be used and called for use with any interface.

6) SEO friendly Development platform: Using this platform, it is very easy to develop SEO-friendly URLs to generate more visits from a specific application.

5b) There is a huge difference between two programs. First program will check condition initially. As the value *$x* is not greater or equal to 5. So, loop will not run. No output will be shown. For second programs do will print result first and then it will check condition. After printing 4 the program will stop. It will print 4 as output.

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

5b).

```
<!DOCTYPE html>
<html lang="en">
<head>
	<meta charset="UTF-8">
	<title>HTML Design</title>
	<style>
		*{margin: 0; padding: 0}
		.wrapper{
			width: 960px;
			margin: 50px auto;
			border: 1px solid #000;
		}
		header{
			min-height: 50px;
			border-bottom: 1px solid #000;
		}
		.content{
			min-height: 302px;
			border-bottom: 1px solid #000;
		}
		.left{
			width: 20%;
			float: left;
			border-right: 1px solid #000;
			min-height: 302px;
		}
		.left ul.menu{
			list-style: none;
			min-height: 100px;
			padding: 10px;
		}
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```
.left ul li{
      border: 1px solid #000;
      padding: 4px;
}
.left ul li a{
      text-decoration: none;
      padding: 0px 10px;
      color: #000;
}
.middle{
      width: 50%;
      float: left;
      padding: 3px;
      border-right: 1px solid #000;
      min-height: 302px;
}
form{
      width: 90%;
      margin: 0px auto;
}
input[type=text] {
      width: 200px;
}
input[type=password]{
      margin-top: 2px;
      width: 200px;
}
button{
      margin-left: 400px;
}
.right{
      width: 25%;
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```
            float: right;
        }
        .right-menu{
            list-style: none;
            padding: 5px;
            width: 98px;
        }
        .right-menu li{
            border: 1px solid #000;
            padding: 5px;
        }
        .right-menu li a{
            text-decoration: none;
            color: #000;
        }
        .center{
            text-align: center;
        }
        footer{
            text-align: center;
            min-height: 50px;
            font-size: 20px;
        }
    </style>
</head>
<body>
    <div class="wrapper">
        <header>
            <h1 class="center">Header Bar</h1>
        </header>
        <div class="content">
            <div class="left">
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```
                    <h3>Left Block</h3>
                    <p class="center">Menu</p>
                    <ul class="menu">
                        <li><a href="#">Home</a></li>
                        <li><a href="#">Registration</a></li>
                        <li><a href="#">About Us</a></li>
                    </ul>
                </div>
                <div class="middle">
                    <h3 class="center">Registration From</h3>
                    <form action="">
                        <label for="">Username</label>
                        <input type="text" name="username"><br>
                        <label for="">Password</label>
                        <input type="password"
name="password"><br>
                        <button type="submit" name="submit"
value="submit">Submit</button>
                    </form>
                </div>
                <div class="right">
                    <img src="something.jpg" alt="">
                    <h3 class="center">Links</h3>
                    <ul class="right-menu">
                        <li><a
href="https://facebook.com">Facebook</a></li>
                        <li><a
href="https://twitter.com">Twitter</a></li>
                        <li><a
href="https://instagram.com">Instagram</a></li>
                        <li><a
href="https://linkedin.com">Linkedin</a></li>
                    </ul>
                </div>
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```
            </div>

            <footer class="">
                  <h3>Footer</h3>
            </footer>
      </div>
      <p class="center">Fig: HTML Layout</p>
</body>
</html>
```


Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741