ourdiu.com | Shared by Unknown_Guy

# Daffodil International University

**Department of Software Engineering**
**Faculty of Science & Information Technology**
**Final Examination, Summer 2025**

**Course Code: SE321; Course Title: Software Engineering Web Application**
**Sections: A,B,C,D,E,F,G,H,I & Teachers: PAS,DB,MI,MMMH,AG,AAM,AMR**

**Time: 2:00 Hrs** **Marks: 40**

## Answer ALL Questions

***[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]***

| | | | | |
|---|---|---|---|---|
| **1.** | *a)* | **Demonstrate** various types of event listeners by developing a JavaScript program, providing appropriate examples, and explaining the functionality of each listener along with how it responds to user interactions. | [Marks-5] | **CLO-2** *Level-3* |
| **2.** | *a)* | <!DOCTYPE html><br><html lang="en"><br><head><br><meta charset="UTF-8"><br><title>Image Click Event</title><br></head><br><body><br><button class="btn">First Button</button><br><img id="myImage" src="image1.jpg" alt="Click me" width="200"><br> Click Count: 0 <br><br><script> // JavaScript code will go here </script><br></body><br></html><br><br>**Construct** a JavaScript function that attaches a click event listener to the image with the ID **myImage**. When the image is clicked:<br>i. Change the image source from "image1.jpg" to "image2.jpg"<br>ii. Increment the number shown in the #count span by 1 each time the image is clicked. | [Marks-5] | **CLO-3** *Level-6* |
| | *b)* | Given two arrays of numbers, write a PHP program to find the **smallest number from the first array** and the **largest number from the second array**, then calculate and return the **average of these two numbers**.<br><br><?php<br>**function averageOfExtremes**($arr1, $arr2) {<br>// Solution code will go here<br>}<br>// Example Usage<br>$arr1 = [4, 1, 2, 1, 2];<br>$arr2 = [2, 2, 3, 3, 5];<br><br>**echo** averageOfExtremes($arr1, $arr2); // Output: 3<br>?><br><br>**Prepare** the solution code by using PHP. | [Marks-5] | |

ourdiu.com | Shared by Unknown_Guy

| | | | | |
|---|---|---|---|---|
| 3. | *a)* | **Design** a PHP OOP program that models a number addition system using an interface and a trait.<br>i. Create an **interface Input** with a method **setNumbers($a, $b)** to set two numbers.<br>ii. Create a **trait Logger** with a method log($message) to display messages.<br>iii. Create a class **Addition** that implements the **Input** interface and uses the **Logger** trait. The class should calculate the sum of the two numbers. Instantiate the class, set two numbers, perform the addition, and print the result along with logs. | [Marks-10] | CLO-4<br>*Level-6* |
| | *b)* | In a Laravel application, the following route is defined in **web.php**:<br>`Route::get('/product', [ProductController::class, 'index'])->name('product.index');`<br>The **ProductController** is located in app/Http/Controllers/ and contains:<br>`public function index()`<br>`{`<br>`  $products = Product::all();`<br>`  return view('products.index', compact('products'));`<br>`}`<br>The Product model is located in **app/Models/** and the corresponding view file is **resources/views/products/index.blade.php**.<br>**Trace** the role and responsibilities of each component in the context of how a browser request to **/product** is handled. | [Marks-8] | |
| | *c)* | In a Student Management System, the administrator should be able to manage student records.<br>**Develop** a PHP-MySQL application to add new students.<br>**Requirements:**<br>• Establish a connection to the MySQL database.<br>• Create a PHP form to enter a student's **name**, **roll number**, and **department**.<br>• Insert the student record into the database upon submission.<br>• Display a confirmation message after successful insertion. | [Marks-7] | |

*Note: You may add or delete rows as per your requirements. Mention CLO's in the right side of the table as per your course outline.*