



Daffodil International University

Faculty of Science & Information Technology

Department of Software Engineering

Final Examination, Fall 2025

Course Code: SE235; Course Title: Desktop and Web Programming
Sections: PAS(C) MHN(A, B, F) AG(D, E) IJC(G, H) ZNM(I, J); Teachers: All
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	a)	Your development team has been assigned to redesign an outdated hospital appointment-management system using the .NET framework. The existing system suffers from poor separation of logic, making updates extremely risky and time-consuming.	[Marks-5]	CLO-2 Level-3
		In this context, illustrate how the MVC (Model-View-Controller) design pattern works within the .NET framework, and describe the role and responsibilities of each component		
✓ 2.	a)	<pre><!DOCTYPE html> <html> <head> <title>Product Rating</title> </head> <body> <div class="item"> <p>Laptop Pro X200</p> <button class="rate-btn">Rate Product</button> </div> <div class="item"> <p>Noise Cancelling Headset</p> <button class="rate-btn">Rate Product</button> </div> <script> // TODO: Add JavaScript code here </script> </body> </html></pre> Develop a JavaScript function that attaches a click event listener to all <p> elements with the <u>class</u> text. When any paragraph is clicked: - Change the font weight of the clicked paragraph to bold. - Log the message "Paragraph clicked!" in the console.	[Marks-5]	CLO-3 Level-6
✓ b)		You are given a base JavaScript object: <pre>const userProfile = { userId: 101, username: "GuestUser",</pre>	[Marks-5]	

	<pre>preferences: [], };</pre> Reconstruct this module to create an array <code>allUsers</code> containing 3 user objects, where each object must include: • <code>userId</code> (unique number) • <code>username</code> (any name) • <code>preferences</code> (array of at least 2 items such as "darkMode", "compactView", etc.		
3. a)	<pre><?php class Account{ protected \$balance; public function __construct(\$balance = 0) { \$this->balance = \$balance; } // TODO: Implement deposit() } class SavingsAccount extends Account { private \$interestRate; public function __construct(\$balance, \$rate) { parent::__construct(\$balance); \$this->interestRate = \$rate; } // TODO: Override a method to include interest in the final balance } ?></pre> Tasks: 1. Implement <u><code>deposit(amount)</code></u> and <u><code>withdraw(amount)</code></u> inside <code>Account</code>. 2. Override a method in <code>SavingsAccount</code> so that it calculates the final balance including interest. 3. Create an object of <code>SavingsAccount</code> with a starting balance of <u>1000</u> and interest rate <u>0.05</u>, call <code>deposit</code> then print the final balance.	[Marks-10]	CLO-4 Level-6
b)	<pre><?php function countCharacter(\$str) { // TODO: Return Odd or Even depending on the number of character } \$input = "DesktopAndWeb"; echo countCharacterFrequency(\$input); ?></pre> Construct the solution so that the function returns an associative array indicating how many times each character appears in the string.	[Marks-5]	

✓ 9)	Below is a defective PHP + MySQL module for managing a Movie Database. It should allow an admin to add and delete movies, but contains logical issues and missing lines. <pre><?php \$connection = mysqli_connect("localhost", "root", "", "moviesdb"); // TODO: Missing validation & SQL handling if(isset(\$_POST['add'])) { \$title = \$_POST['title']; \$year = \$_POST['year']; // INSERT query missing } if(isset(\$_POST['delete'])) { \$movieId = \$_POST['movieId']; // DELETE query missing } ?> <form method="POST"> <h3>Add Movie</h3> <input type="text" name="title" placeholder="Title">
 <input type="number" name="year" placeholder="Year">
 <button name="add">Add Movie</button> <h3>Delete Movie</h3> <input type="number" name="movieId" placeholder="Movie ID">
 <button name="delete">Delete Movie</button> </form></pre> Construct a correct and complete PHP application that: 1. Performs INSERT and DELETE operations. 2. Validates that fields are not empty. 3. Displays meaningful success/error messages. 4. Uses proper MySQL query execution.	[Marks-10]	
-------------	---	-------------------	--