Daffodil International University
Faculty of Science & Information Technology
Department of Software Engineering
Final Examination, Summer 2026
Course Code: SE 235 Course Title: Desktop and Web Programming
Sections & Teachers: 43 {A, B, O(PC), C, D, E(TT), F, G(KM), H(SAC), I(SSI), K, L(RUA), J, M, N(PS)}

Time: 2 Hour **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 university is developing a simple Course Registration System using the Node.js http module. A student submits a registration form from the browser containing the following information as a request body in a POST request: **sid=123&course=se235&department=SWE**			
	a)	Apply Node.js HTTP request-handling concepts to write JavaScript code that handles the incoming HTTP request using the data and end events . Collect the complete request body and convert the received form data into a JavaScript object so that the values of sid, course, and department can be accessed individually.	5	CLO -3 Level -3
	b)	Apply Node.js response-handling concepts to generate the following response **dynamically** from the submitted form data: **Hello 123, welcome to se235 in the department of SWE.** Your code should generate the message dynamically from the submitted form data, rather than hard-coding the values.	5	
	c)	Implement the ElectronJS IPC communication described below. An ElectronJS desktop application has a "Show Notification" button in the renderer process. When the user clicks the button, the renderer should invoke an IPC channel and send a notification message to the main process. The main process should receive the data, display it using an Electron dialog box, and return a confirmation message to the renderer. Write down the necessary code to implement the feature described above: - Define (handle) a signal from the main process that shows the notification. - Invoke (emit) the signal from a renderer process when appropriate. Assume appropriate preloader, explain your assumption if you use any preloader features	5	

	d)	Apply Node.js file-system concepts to write JavaScript functions that: (i) save the following string in a text file named student.txt; `const data = "Hello my name is Rahim"` (ii) Read the contents of student.txt and return the data as a UTF-8 string.	5	
2	a)	Show the Git workflow to demonstrate how changes move through the three main sections of a Git project: the working tree, the staging area, and the Git repository. Show the relevant Git commands in the correct sequence.	5	CLO -4 *Level -3*
	b)	Apply your understanding of npm to demonstrate how a software engineer can use NPM to obtain and manage a project dependency. Show the command used to install ElectronJS in a project and describe the significance of package.json.	5	
	c)	Apply JavaScript object concepts to create an object named customer containing customerId, restaurantName, foodItem, and quantity. Assign appropriate sample values to the object. Then, convert the object into JSON format and store it in a variable. Afterwards, Write JavaScript statements to display the JSON data containing the variable and its data type. What will be shown in the console after you execute this code?	5	
	d)	Apply ElectronJS desktop application concepts to write code that creates a BrowserWindow with a starting size of 1000 × 700 (width × height) pixels, a minimum size of 600 × 400 pixels, the title "DIU Desktop App", a center-aligned position, and index.html loaded into the window.	5	