Daffodil International University

Faculty of Science & Information Technology

Department of Computing and Information System

Final Examination, Summer-2026

Course Code: CIS235, Course Title: Software Engineering

Level: 2 Term: 3

Exam Duration: 2 Hours **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)*	Explain the key phases of the Software Development Life Cycle (SDLC) and discuss why documentation is essential across all development phases.	[4]	CO 1
	b)	Differentiate between Functional Requirements (FRs) and Non-Functional Requirements (NFRs).	[2]	
	c)	A healthcare organization is building a Mobile Patient Health Portal. Apply your knowledge of requirements classification to construct two Functional Requirements and two Non-Functional Requirements for this platform, ensuring each written statement is verifiable and unambiguous	[3]	
2.		OmniCart is an online shopping platform running on a single, large codebase (a monolithic Java application) that handles orders, inventory, users, and payments all together. During big sales events, the application frequently crashes. The chief architect found that when the inventory module updates stock levels, it directly changes global variables that belong to the payment module. To fix these performance and reliability issues, the team wants to modularize the system and is considering moving to a microservices architecture. i. Based on the scenario, identify the current type of coupling and cohesion present in OmniCart's codebase. ii. Explain two major software engineering risks caused by this current setup. iii. Propose how the engineering team can refactor the code to achieve Functional Cohesion and Data Coupling. iv. Compare a Layered Architecture with a Microservices Architecture for OmniCart in	[5x2 = 10]	CO 2

		terms of system maintainability versus performance. v. Using the ATAM (Architecture Tradeoff Analysis Method) framework, identify and briefly explain: ■ One Sensitivity Point ■ One Trade-off Point		
3.	*a)*	Explain the Single Responsibility Principle (SRP) and the Open/Closed Principle (OCP) in low-level object-oriented design.	[2]	CO 3
	b)	A developer has built an InvoiceManager class that calculates tax formulas, prints invoice PDF layouts, and saves records directly to a database. Explain why this design violates SRP. Show how to refactor this structure into separate classes.	[4]	
	c)	Define Code Refactoring and state its fundamental rule regarding external system behavior	[2]	
	d)	Explain how Docker containerization solves the "works on my machine" deployment issue	[2]	
4.	*a)*	Analyze how neglecting Preventive Maintenance leads to structural degradation under Lehman's *Law of Increasing Complexity*. Relate this degradation to the "Software Cost Iceberg" concept, where 60% to 80% of total lifecycle costs reside below the waterline	[4]	CO 4
	b)	Contrast Verification ("Are we building it right?") with Validation ("Are we building the right thing?")	[2]	

	c)	Analyze the following conditional code structure and calculate its Cyclomatic Complexity using Thomas McCabe's decision formula V(G)=D+1
		
		`void processTransaction(int amount, bool isVerified)`
		`{`
		`  if (amount > 5000 && isVerified == true) {`
		`  flagHighValue();`
		`  } else {`
		`  standardProcess();`
		`  }`
		`}`
			[4]	