ourdiu.com | Shared by Md. Mahid Alam Marin

Daffodil International University

Faculty of Science & Information Technology
Department of Computing and Information System
Final Examination, Spring-2026
Course Code: CIS-216, Course Title: OBJECT ORIENTED PROGRAMMING
Level: 2 Term: 1

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.]

Welcome to "NextGen Tech"

You have just been hired as a Junior Software Architect at NextGen Tech, a rapidly growing startup that is building an "All-in-One E-Commerce & Delivery Platform." Currently, their software is old, buggy, and struggling to handle new users. Over the next two hours, you must guide the company through a complete software rewrite, designing new features, fixing crashes, and building modern web APIs.

1.		On your first day, the CEO tells you that NextGen Tech's original software was written years ago using procedural programming (C-style). It is now a messy "spaghetti code" nightmare. You propose rewriting the entire platform in Java using Object-Oriented Programming (OOP).		CO1
	a)	The CEO asks, "Why should we spend money shifting from procedural to OOP?" Explain the evolution of modular programming into OOP. Give two specific reasons why OOP is vastly superior for managing a massive E-Commerce platform.	4	L2
	b)	To prove your point, explain how you will use the four pillars of OOP (Encapsulation, Inheritance, Polymorphism, Abstraction) in the new system. Provide one brief example of how each pillar will be applied to the E-Commerce platform (e.g., What data will be encapsulated? Who inherits from whom?).	4	L3
2.		The rewrite is approved! Your first coding task is to build a new Payment Gateway. NextGen Tech wants customers to be able to pay using multiple methods: Credit Card, PayPal, and Crypto.		CO2
	a)	You need to establish the blueprint for these payments. Differentiate between an **Interface** and an **Abstract Class**. Explain to your junior developers what "100% abstraction" means in Java, and state the rules regarding constructors for both Interfaces and Abstract Classes.	4	L3
	b)	Write a brief Java code snippet demonstrating **Polymorphism (Method Overriding)** for this feature. Create an interface named Payment with an abstract method processPayment(), and implement it in a PayPal subclass.	4	L3
3.		It is "Black Friday," the busiest shopping day of the year. Millions of users are placing orders simultaneously. The system is crashing because it processes orders one by one, and some user data is corrupted.		CO4
	a)	To fix the speed issue, you decide to implement **Threads**. Explain what a Thread is and outline its **Life Cycle**. Why will using threads help NextGen Tech survive the Black Friday traffic?	4	L4

ourdiu.com | Shared by Md. Mahid Alam Marin

	b)	To fix the crashing issue, write a Java code snippet using the **Collection Framework** (e.g., an ArrayList) to store incoming OrderIDs. Wrap your collection logic in an **Exception Handling** block (try, catch, finally) to safely catch a potential IndexOutOfBoundsException caused by corrupted data. In the comments of your code, briefly explain the difference between the throw and throws keywords.	4	L4
4.		With payments working, the company is launching a new "Same-Day Food Delivery" service. Before you are allowed to write any Java code for this feature, the Lead Architect demands a UML diagram.		CO3
	a)	Explain the purpose of a **UML Diagram**. Why is it a necessary step before typing code?	3	L2
	b)	Draw a **Class Diagram** for this new delivery feature. You must include at least three interacting classes. For each class, list at least two properties (variables) and two methods, and indicate the **relationships** between the classes.	5	L5
5.		NextGen Tech is now a massive success. To allow third-party sellers to manage their own products via a mobile app, you are tasked with building a modern RESTful Web API using **Spring Boot**.		CO4
	a)	Spring Boot requires a strict layered architecture to keep the code organized. Describe the specific roles of the **Model class, Repository layer, Service class**, and **Controller layer** in managing the new "Seller Product Data."	4	L4
	b)	Explain how the Controller handles incoming network requests. Differentiate between the @GetMapping and @PostMapping annotations. Provide a simple 2-line code example showing how to use one of these annotations to fetch or save a product in your controller.	4	L4

@Get

Con