ourdiu.com | Shared by Aftab Hosen

Daffodil International University

Faculty of Science & Information Technology
Department of Computer Science & Engineering
Final Examination, Summer 2026
Course Code: CSE221, Course Title: Object Oriented Programming
Level:2 Term:1 Batch:68

Time: 02:00 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)*	What is a Java Collection? Mention two advantages of using the Java Collections Framework.	3	CO2
	b)	What is a Map? Explain the meaning of key-value pair with one example.	3	
	c)	Explain why ArrayList is suitable for maintaining an ordered list of student records. Mention whether it allows duplicate elements.	4	
2.		A Vehicle Service Center manages maintenance tasks for different types of vehicles, specifically Cars and Bikes. All vehicles must support two distinct tiers of maintenance: basic servicing (such as oil changes) and advanced servicing (such as engine tuning). To standardize operations, the system requires two interfaces: ServiceA containing the method basicService() and ServiceB containing the method advancedService(). Both Car and Bike classes must implement both interfaces to provide their respective service behavior.		CO2
	a)	Draw a UML class diagram representing the system design.	5	
	b)	Implement a Java program that includes the interfaces and classes according to the specification. In your main method, instantiate one Car and one Bike object, and execute both service methods on each object.	5	
3.		In an e-commerce system, a PaymentProcessor class needs to record transaction details for auditing. To do this, it uses a Logger object — but only briefly, inside a single method that calls Logger's method to write the log entry. The PaymentProcessor does not keep the Logger as one of its fields, and the Logger object is not created or stored beforehand; it is simply passed in and used at the moment it is needed, then discarded.		CO3
	a)	Relate which UML relationship exists between PaymentProcessor and Logger, illustrate UML Class diagram. Justify your answer.	5	
	b)	Develop a Java code for both classes, showing how PaymentProcessor uses Logger inside a method to log a transaction.	5	
4.		Imagine you are developing the core payment module for an online food delivery application. When a user places an order, the application attempts to process the transaction by deducting the order total from the user's digital wallet balance. Before proceeding with the transaction, the payment module must check for two critical conditions: the payment amount must be strictly greater than zero, and the user's account must have sufficient funds to cover the purchase. If either condition fails, the system must immediately generate a specific runtime exception with an explanatory error message to halt invalid processing. Additionally, during this payment operation, a network socket connection to the bank's gateway server is		CO4

ourdiu.com | Shared by Aftab Hosen

		opened. Because network resources are limited, the system must ensure that the connection to the gateway server is always closed and logged, whether the transaction finishes successfully, encounters a validation error, or experiences an unexpected failure.		
	a)	Based on the scenario above, write a Java program containing a method processPayment(double balance, double amount) and a main method to simulate this process. Your code must explicitly handle input validation using the ***throw*** keyword and manage transaction flow and cleanup using **try**, **catch**, and **finally** blocks.	*6*	
	b)	Explain the technical roles of **throw** and **finally** in your Java solution. Why is the **finally** block critical in this payment gateway scenario?	4	

ourdiu.com

ourdiu.com | Shared by Aftab Hosen

Daffodil International University

Faculty of Science & Information Technology
Department of Computer Science & Engineering
Final Examination, Summer 2026
Course Code: CSE221, Course Title: Object Oriented Programming
Level:2 Term:1 Batch:68

Time: 02:00 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)*	What is a Java Collection? Mention two advantages of using the Java Collections Framework.	3	CO2
	b)	What is a Map? Explain the meaning of key-value pair with one example.	3	
	c)	Explain why ArrayList is suitable for maintaining an ordered list of student **records**. Mention whether it allows duplicate elements.	4	
2.		A Vehicle Service Center manages maintenance tasks for different types of vehicles, specifically Cars and Bikes. All vehicles must support two distinct tiers of maintenance: basic servicing (such as oil changes) and advanced servicing (such as engine tuning). To standardize operations, the system requires two interfaces: ServiceA containing the method basicService() and ServiceB containing the method advancedService(). Both Car and Bike classes must implement both interfaces to provide their respective service behavior.		CO2
	a)	Draw a UML class diagram representing the system design.	5	
	b)	Implement a Java program that includes the interfaces and classes according to the specification. In your main method, instantiate one Car and one Bike object, and execute both service methods on each object.	5	
3.		In an e-commerce system, a PaymentProcessor class needs to record transaction details for auditing. To do this, it uses a Logger object — but only briefly, inside a single method that calls Logger's method to write the log entry. The PaymentProcessor does not keep the Logger as one of its fields, and the Logger object is not created or stored beforehand; it is simply passed in and used at the moment it is needed, then discarded.		CO3
	a)	Relate which UML relationship exists between PaymentProcessor and Logger, illustrate UML Class diagram. Justify your answer.	5	
	b)	Develop a Java code for both classes, showing how PaymentProcessor uses Logger inside a method to log a transaction.	5	
4.		Imagine you are developing the core payment module for an online food delivery application. When a user places an order, the application attempts to process the transaction by deducting the order total from the user's digital wallet balance. Before proceeding with the transaction, the payment module must check for two critical conditions: the payment amount must be strictly greater than zero, and the user's account must have sufficient funds to cover the purchase. If either condition fails, the system must immediately generate a specific runtime exception with an explanatory error message to halt invalid processing. Additionally, during this payment operation, a network socket connection to the bank's gateway server is		CO4

ourdiu.com | Shared by Aftab Hosen

		opened. Because network resources are limited, the system must ensure that the connection to the gateway server is always closed and logged, whether the transaction finishes successfully, encounters a validation error, or experiences an unexpected failure.		
	a)	Based on the scenario above, write a Java program containing a method processPayment(double balance, double amount) and a main method to simulate this process. Your code must explicitly handle input validation using the **throw** keyword and manage transaction flow and cleanup using **try**, **catch**, and **finally** blocks.	*6*	
	b)	Explain the technical roles of **throw** and **finally** in your Java solution. Why is the **finally** block critical in this payment gateway scenario?	4	

ourdiu.com