



Daffodil International University  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Final Examination, Spring 2025

**Course Code: CSE221, Course Title: Object Oriented Programming**

**Level:2 Term:2 Batch: ALL**

**Time: 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 concept of <b>Inheritance, Abstract Class, and Polymorphism</b> in Java. How do they work together to create reusable and maintainable code?	5	CO1
	b)	<p><b>Scenario:</b> A company manages its employees using an <b>Employee Management System</b>. The system consists of:</p> <ul style="list-style-type: none"><li>• <b>Base Class (Employee)</b> with attributes: name, ID, salary.</li><li>• <b>Abstract Class (PermanentEmployee)</b> which extends Employee and has additional attributes: bonus.</li><li>• <b>Concrete Class (ContractEmployee)</b> which extends Employee and has an attribute contractPeriod.</li><li>• Implement <b>Polymorphism</b> by overriding a method calculateSalary() for both PermanentEmployee and ContractEmployee.</li></ul> <p><b>Task:</b></p> <ol style="list-style-type: none"><li>I. Implement the <b>Inheritance &amp; Abstract Class</b> structure in Java. (6)</li><li>II. Implement calculateSalary() method using polymorphism. (2)</li><li>III. Demonstrate the use of <b>method overriding</b>. (2)</li></ol> <p>(Full correct implementation: 10 marks, Partial correctness: 6-8 marks, Minor issues: 3-4 marks)</p>	10	CO2
2.	a)	<p>How does Java achieve multiple inheritance using Interfaces?</p> <ol style="list-style-type: none"><li>I. Explain with an example how an interface can be used to achieve multiple inheritance in Java. (3)</li><li>II. Discuss <b>advantages and limitations</b> of using interfaces for multiple inheritance. (2)</li></ol>	5	CO1

	<p>b) <b>Scenario:</b>  <b>A Library System</b> consists of:</p> <ul style="list-style-type: none"> <li>• A Library that manages multiple Books.</li> <li>• Each Book has an Author and belongs to a Category.</li> <li>• Each Book is issued to a Member.</li> </ul> <p><b>Task:</b></p> <ol style="list-style-type: none"> <li>I. Design a <b>UML Class Diagram</b> covering <b>Inheritance, Abstract Class, Polymorphism, and Association</b> between Library, Book, Author, Category, and Member. (6)</li> <li>II. Clearly define relationships such as <b>One-to-Many, Many-to-Many</b>. (2)</li> <li>III. Justify the use of <b>Abstract Classes</b> and <b>Interfaces</b> if applicable. (2)</li> </ol> <p><i>(Full correct implementation: 10 marks, Partial correctness: 6-8 marks, Minor issues: 3-4 marks)</i></p>	10	CO3
3.	<p><b>Scenario:</b>  <b>A Ride-Sharing Application</b> needs a system to manage:</p> <ul style="list-style-type: none"> <li>• <b>Drivers (Driver class)</b></li> <li>• <b>Passengers (Passenger class)</b></li> <li>• <b>Rides (Ride class)</b></li> <li>• <b>Vehicles (Vehicle class)</b></li> </ul> <p>The system should:</p> <ul style="list-style-type: none"> <li>• Allow a Passenger to book a Ride.</li> <li>• Assign an available Driver to the ride.</li> <li>• Ensure each Ride is associated with a Vehicle and a Driver.</li> <li>• Implement calculateFare() as an <b>abstract method</b> that varies based on ride type (Economy, Premium).</li> </ul> <p><b>Task:</b></p> <ol style="list-style-type: none"> <li>a) <b>Design a UML Class Diagram</b> for the system. (7)</li> <li>b) <b>Identify Associations</b> (One-to-Many, Many-to-Many). (3)</li> </ol>	10	CO3