

Code: AEIOU

Batch: _____

ID: _____



Daffodil International University
Faculty of Science & Information Technology
Department of Information Technology & Management
Final Examination, Summer 2025

Course Code: ITM213; Course Title: Object Oriented Concepts
Sections & Teachers: MA

Time: 2:00 Hrs**Marks: 40**

Answer ALL Questions

In a Supershop Management System, the base class **Person** is introduced to represent individuals interacting with the system. It contains two attributes (**name**, **contact**) and two methods (**displayInfo()**, **updateInfo()**), and serves as a parent class for **Customer** and **Employee**. Through **inheritance**, both **Customer** and **Employee** override the **displayInfo()** method to present information specific to their roles. The **Customer** class adds attributes like **customerId** and **loyaltyPoints**, with methods **addPoints()** and **redeemPoints()**. The **Employee** class, through **constructor chaining**, introduces attributes like **employeeId** and **role**, along with methods **assignTask()** and **calculateSalary()**. Employees are further specialized into **Cashier** and **Manager**, illustrating deeper inheritance.

The **Product** class represents items in the store, with attributes (**productCode**, **price**) and methods (**getDetails()**, **applyDiscount()**). It is linked to the **Category** class, where each product must belong to exactly one category (e.g., **Grocery**, **Electronics**), and a category cannot exist independently of its products. On the other hand, a **ShoppingCart** class demonstrates holding multiple **Product** objects temporarily, while still allowing those products to exist independently of the cart.

The **Bill** class shows association with both **Customer** and **cashier** — a bill is generated by a cashier for a particular customer. It contains attributes (**billId**, **totalAmount**) and methods (**generateBill()**, **printBill()**). To handle flexible billing operations, the **Bill** class includes an overloaded method **addItem()**, which has three different implementations: one version accepts only a **Product**, another accepts a **Product** with a quantity, and a third accepts a **Product**, quantity, and a discount percentage and prints the details.

Every class provides **getter** and **setter** methods for controlled access to attributes, applying appropriate access modifiers.

1.	a)	Design an UML class diagram for the above scenario. Design the classes and relationship among the classes properly and justify your choice.	[Marks-10]	CO-4 Level-6
2.	a)	Implement the Person , customer and Employee classes in python. Implement constructor chaining, method overriding	[Marks-5]	CO-3 Level-3

Code: AEIOU		Batch:	ID:	
		properly. Create objects of each class, call the methods and show the output		
	b)	Consider the Bill class. Implement method overloading using dispatcher module.	[Marks-5]	CO-2 Level-3
	c)	Consider the Product class, implement it as an abstract class. Derive a class "S_product" from the abstract class with specific method implementation.	[Marks-5]	CO-3 Level-3
3.	a)	Differentiate between method overloading and method overriding in your own words. Explain the role of method resolution order (MRO) in method overriding concept.	[Marks-5]	CO-1 Level-4
	b)	Define polymorphism and encapsulation in your own words. Explain different types of access modifiers with scope of each access modifiers	[Marks-5]	CO-1 Level-2
4.	a)	In a Student Result Management System , the program stores student marks in a list. When a user tries to access the marks of a student by providing an <u>index</u> , there is a possibility that the entered index is outside the <u>valid range of the list</u> . To handle this situation, the system applies a try block for accessing the mark, an except block to catch invalid index access, an else block to display the mark when the <u>index is valid</u> , and a finally block to show a closing message like "Operation Completed. Closing Result System". Now implement the above scenario in python.	[Marks-5]	CO-3 Level-3