U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

January / February 2025 Semester End Main Examinations

Programme: B.E. **Semester: III**
Branch: Artificial Intelligence & Machine Learning **Duration: 3 hrs.**
Course Code: 23AM3PCOOP **Max Marks: 100**
Course: Object Oriented Programming

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

			UNIT - I	*CO*	*PO*	Marks
1	a)		Illustrate the principles of Object-Oriented Programming with an example.	*CO1*	*PO1*	**07**
	b)		How do programming environments differ across different languages and platforms? Explain	*CO1*	*PO2*	**06**
	c)		What is an abstract machine, and how does it relate to programming languages?	*CO1*	*PO1*	**07**
			OR			
2	a)		Compare and contrast imperative, functional, and object-oriented programming paradigms with examples.	*CO1*	*PO2*	**07**
	b)		Explain the role of Integrated Development Environments (IDEs) in modern software development. Provide examples.	*CO1*	*PO1*	**06**
	c)		Describe the functioning of the interpreter for a generic abstract machine with a neat diagram.	*CO1*	*PO1*	**07**
			UNIT - II			
3	a)		Create a full working python function that satisfies the following requirements (following indentation rules properly) i. Create an outer function that will accept two parameters, a and b. ii. Create an inner function inside an outer function that will calculate the product of a and b. iii. At last, an outer function will add 2 to the product and return it.	*CO2*	*PO3*	**08**
	b)		Design a class representing a library catalog. Encapsulate attributes such as book ID, title, author, and availability status. Implement methods to check out books, return books, and search for books by title or author. Ensure that book availability is updated correctly and that sensitive information like borrower details is handled securely.	*CO2*	*PO3*	**08**

		c)	Differentiate instance method and class method. Illustrate each one with a suitable example.	*CO2*	*PO2*	**04**
			OR			
	4	a)	Analyze the following description and implement python code using both parameterized and non-parameterized constructors. Class: Employee Objects: Manager, CEO, Fin_Head Data Members: employee_id, department, date of joining, salary Member functions: on_duty, get_pay, availability.	*CO2*	*PO3*	**08**
		b)	Create a class to represent student information. Encapsulate attributes such as student ID, name, course enrolled, and grades. Provide methods to update grades, calculate GPA, and display student information. Ensure that sensitive data like grades is only accessible through appropriate methods and access controls.	*CO3*	*PO3*	**08**
		c)	Given the following two data, using anonymous functions get the desired output. *numbers = [1, 3, 4, 5, 6, 7, 8,9], filter= [1,4,7,9]* *Expected O/p: [2,3,5,6,8]*	*CO2*	*PO2*	**04**
			UNIT - III			
	5	a)	Implement a Python program for managing employee records. Create a base class Employee with attributes like name, employee_id, and methods like calculate_salary(). Then, create subclasses like Manager, Developer, and HR, each inheriting from Employee and adding specific functionalities for their roles.	*CO3*	*PO3*	**08**
		b)	Create a Python class Animal with a method make_sound(). Then, create subclasses like Dog, Cat, and Bird, each inheriting from Animal. Override the make_sound() method in each subclass to represent the sounds made by each animal.	*CO3*	*PO3*	**08**
		c)	Illustrate 'is-a' relation of OOPs principle with one example.	*CO2*	*PO1*	**04**
			OR			
	6	a)	Describe the concept of Diamond Problem in multiple inheritance with suitable example	*CO2*	*PO3*	**05**
		b)	Implement an abstract base class BankAccount in Python representing a bank account. Define abstract methods deposit() and withdraw(). Create concrete subclasses like SavingsAccount and CheckingAccount that inherit from BankAccount and implement these methods according to their specific account types.	*CO3*	*PO3*	**10**
		c)	How do you access parent class in child class? Provide your answer with an example.	*CO2*	*PO1*	**05**
			UNIT - IV			
	7	a)	i. Differentiate error & exception.	*CO2*	*PO2*	**07**

			ii. Anticipate what type of valid errors can be handled in a proper order. **try:** x = 1 / "a" y = open("file_not_exist.txt") z = "a" + None **except**(Fix Code here) as Fix Code here: print("Error:", Fix Code here) x, y, z = None, None, None print("x:", x) print("y:", y) print("z:", z)			
		b)	Explain the try-except block and analyze how it helps in catching and handling exceptions with an example	***CO2***	***PO3***	**07**
		c)	How the concept of polymorphism is implemented in object-oriented programming. Demonstrate with an example.	***CO2***	***PO2***	**06**
			OR			
	8	a)	Evaluate the role of polymorphism in reducing code duplication. Provide examples to support your argument.	***CO2***	***PO2***	**06**
		b)	Create a base class called Father and derived class called Son which extends the base class. In Father class, implement a constructor which takes the age and throws the user defined exception WrongAge( ) when the input age=father's age.	***CO1***	***PO3***	**07**
		c)	Create a Python program to show polymorphism with classes: i. Define a base class Shape with a method area() that returns 0. ii. Create two derived classes, Rectangle and Circle, that override the area() method to calculate and return the area of the respective shapes.	***CO2***	***PO2***	**07**
			UNIT - V			
	9	a)	Develop a code that implements a concurrency mechanism for printing numbers (1 to 5) and alphabets (A to D) in a span of 10 seconds.	***CO2***	***PO3***	**06**
		b)	Explain the benefits of multithreading in terms of concurrency, responsiveness, and resource utilization.	***CO1***	***PO1***	**07**
		c)	Define thread deadlock and analyze how it occurs in multithreaded programs.	***CO1***	***PO2***	**07**
			OR			
	10	a)	Explain the stages in the thread life cycle with suitable diagram.	***CO1***	***PO1***	**06**
		b)	Why is thread synchronization necessary? Write a Python program to demonstrate synchronization using a lock.	***CO1***	***PO2***	**07**
		c)	Illustrate the inter thread communication in brief. Develop python code for inter thread communication using queue.	***CO1***	***PO2***	**07**

******