U.S.N.										

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

Autonomous Institute Affiliated to VTU

June 2025 Semester End Main Examinations

Programme: B.E. **Semester: III**

Branch: Artificial Intelligence and 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)	Design a class Employee that models the following: Attributes: name, employee_id, department, and salary. Define methods to: 1. Initialize the parameters using a constructor. 2. Display employee details. 3. Calculate and display the yearly salary (12 times the monthly salary). Using the Employee class, create instances for at least two employees, set their attributes, and demonstrate calling the methods to display their details and yearly salary.	*CO 2*	*PO 2*	**10**
	b)	Describe the functioning of the interpreter for a generic abstract machine with a neat diagram.	*CO 1*	*PO 1*	**10**
		OR			
2	a)	Create a Country class with attributes for name, capital, and population. Instantiate two country objects with diverse attributes.	*CO 2*	*PO 1*	**08**
	b)	Differentiate functional and object-oriented programming languages. Provide relevant code snippets for each.	*CO 1*	*PO 1*	**06**

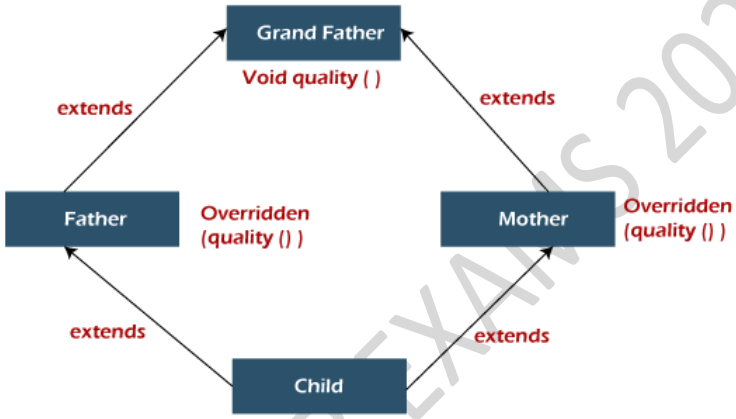
	c)	Predict the output of the given codes and justify your answer.
		i.
		`class Demo:`
		`  def __init__(self):`
		`  name = "Demo"`
		`  def display(self):`
		`  print(name)`
		`d = Demo()`
		`d.display()`
			CO 1	*PO 2*	**06**

			ii.
			class Test:
			def __init__(self):
			self.name = "Python"
			t1 = Test()
			t2 = Test()
			t1.name = "Java"
			print(t2.name)			

			UNIT - II			
	3	a)	Design a Python class named MenuItem to encapsulate information about items on a restaurant menu. Include private attributes for the item name, price, and dietary information. Implement methods for setting and getting these attributes. Also, include a method to determine if the item is vegetarian.	*CO 2*	*PO 2*	**10**
		b)	Explain 3 types of methods in python with working code for each.	*CO 1*	*PO 1*	**10**
			OR			

	4	a)	Complete the following code with relevant type of method.
			Class Rectangle:
			def __init__(self, length, width):
			// fix the code
			def calculate_area(self):
			// fix the code
			def calculate_perimeter(self):
			// fix the code
			rectangle_instance = Rectangle(5, 8)
			area = rectangle_instance.calculate_area()
			perimeter = rectangle_instance.calculate_perimeter()
				CO 2	*PO 2*	**05**

		b)	How would you employ access specifiers in Python to enhance the security and encapsulation of user-related attributes, such as usernames, passwords, and email, in the context of developing a messaging application? Comment on the scope of each access specifier with an example.	*CO 1*	*PO 2*	**10**
		c)	How does anonymous functions benefit over regular functions. Demonstrate lambda functions to find the largest element in a list.	*CO 1*	*PO 1*	**05**
			UNIT - III			
	5	a)	Summarize the concept of multi-level inheritance and demonstrate the same for the following instance: i. Create a base class Animal with attributes like species. ii. Derive a class Mammal from Animal with additional attributes like sound. iii. Further derive classes like Dog and Cat from Mammal with specific methods for barking and meowing.	*CO 2*	*PO 1*	**10**

			iv. Invoke methods of classes Dog and Cat in main function.			
		b)	Design an abstract class Employee with abstract methods for calculating salary and displaying employee details. Derive concrete classes like Manager and Developer from Employee and implement the abstract methods.	*CO 2*	*PO 1*	**10**
			OR			
	6	a)	Consider the scenario given below and answer the following questions: i. Identify the type of inheritance in the scenario given below. ii. Assess the efficiency of the execution of the method 'quality ()' in the 'Child' class. Substantiate your response. iii. Illustrate the approach through which Python addresses the issue identified in question (ii). 	*CO 2*	*PO 1*	**06**
		b)	Illustrate the creation of an Abstract Base Class in Python by providing a concise explanation and a practical example.	*CO 1*	*PO 1*	**06**
		c)	Write a program to demonstrate multiple inheritance: • Define a base class Vehicle with attributes brand and speed. • Create derived classes Car and Bike with additional attributes like num_wheels and methods such as display_details().	*CO 2*	*PO 2*	**08**
			UNIT - IV			
	7	a)	Define polymorphism. Demonstrate the following ways of implementing polymorphism in python with an example for each. i. User defined Method overloading. ii. Overloading in Built in methods. iii. Operator overloading	*CO 2*	*PO 1*	**10**
		b)	Distinguish between bug, error and exception.	*CO 1*	*PO 1*	**05**
		c)	Write a Python program that executes division and handles an ArithmeticError exception if there is an arithmetic error.	*CO 2*	*PO 2*	**05**
			OR			

	8	a)	Illustrate the functions of various blocks in exception handling with an example.	*CO 2*	*PO 1*	**10**
		b)	Implement a class MathOperations with methods for basic arithmetic operations like addition and subtraction. Demonstrate method overloading to handle different types and numbers of operands, allowing users to perform calculations with varying complexities.	*CO 2*	*PO 1*	**10**
			UNIT - V			
	9	a)	Operating systems assigns a thread to take a print-out of a document which is stored in the system hard disk. Identify the different states of a thread with a neat diagram until work done.	*CO 2*	*PO 1*	**07**
		b)	Consider you are developing a Banking system where multiple clients can access their accounts simultaneously. Implement thread synchronization mechanisms to ensure that withdrawals and deposits to the same account are executed safely without encountering race conditions or data inconsistencies.	*CO 2*	*PO 2*	**08**
		c)	Illustrate the inter thread communication in brief.	*CO 2*	*PO 1*	**05**
			OR			
	10	a)	Write a python program to create threads to perform tasks such as printing odd and even numbers between 1 to 50. Terminate the main process only after the completion of the threads created.	*CO 2*	*PO 2*	**08**
		b)	Describe the following with an example for each: i. Daemon threads ii. Thread pools iii. Inter-Thread Communication iv. Thread Deadlock	*CO 2*	*PO 1*	**08**
		c)	Differentiate concurrency and parallelism with an example.	*CO 1*	*PO 1*	**04**

****