

B. TECH DEGREE EXAMINATIONS: NOV/DEC 2024

(Regulation 2018)

Second Semester

ARTIFICIAL INTELLIGENCE & DATA SCIENCE

U18AII2204: Object Oriented Programming and Data Structures

COURSE OUTCOMES**CO1:** Understand the object-oriented programming concepts and GUI.**CO2:** Apply Overloading and the concept of handling exceptions.**CO3:** Demonstrate the concepts of data structures using Python.**CO4:** Develop graph, sorting, and search techniques of data structures.**Time: Three Hours****Maximum Marks: 100****Answer all the Questions****PART A (10 x 2 = 20 Marks)****(Answer not more than 40 words)**

1.	Define the concept of inheritance and its importance in object-oriented programming.	CO1	[K ₁]
2.	Explain the difference between classes and objects in Python.	CO1	[K ₂]
3.	What is encapsulation and give an example.	CO1	[K ₁]
4.	Write the purpose of the try and except blocks in Python exception handling.	CO2	[K ₂]
5.	Differentiate method overloading and method overriding.	CO2	[K ₃]
6.	List two real-world applications of the stack data structure.	CO3	[K ₁]
7.	Identify the purpose of the push() and pop() operations in stack management.	CO3	[K ₁]
8.	Create an expression tree for the expression $((a+(b/c)*d))-e$.	CO3	[K ₃]
9.	Differentiate between depth-first search and breadth-first search.	CO4	[K ₃]
10.	Explain postfix notation and its advantages over infix notation.	CO4	[K ₂]

Answer any FIVE Questions:-**PART B (5 x 16 = 80 Marks)****(Answer not more than 400 words)**

11.	a)	Apply the concept of polymorphism by creating a Python program that demonstrates method overriding.	CO1	[K ₃]
	b)	Create a Python GUI application using Tkinter for a basic calculator that performs addition, subtraction, multiplication, and division.	CO1	[K ₄]
12.	a)	Explain the concept of multiple inheritance with an example.	CO1	[K ₂]

	b)	Construct a simple banking application in Python with classes for Customer and Account. The Customer class should include attributes for name and ID, and the Account class should include methods for deposit and withdrawal. Illustrate how to interact with these classes.		CO1	[K ₃]
13.	a)	Outline the concept of operator overloading with a suitable example		CO2	[K ₂]
	b)	Illustrate error handling by creating a Python program that catches and handles specific exceptions during file operations.		CO2	[K ₃]
14.	a)	Write a Python program to implement a stack using lists. Include methods for push, pop, peek, and is_empty, and demonstrate how the stack could be used to check for balanced parentheses in an expression.		CO3	[K ₃]
	b)	Implement a queue-based approach in Python to simulate a simple task scheduler. Use a queue to manage tasks that arrive in a first-come, first-served order. Each task should have a name and a priority level. Include methods to add tasks and process them based on priority.		CO3	[K ₃]
15.	a)	Write a Python program that uses a stack to convert an infix expression (a + b) * (c - d) to its equivalent postfix expression.		CO3	[K ₃]
	b)	Develop a Python program to implement a doubly linked list and demonstrate how it can be used to store and manage a playlist of songs. Include methods for adding and removing songs, as well as traversing forwards and backwards through the playlist.		CO3	[K ₃]
16.	a)	Implement Dijkstra's algorithm in Python to find the shortest path in a weighted, undirected graph. Provide test cases to demonstrate your implementation and show the step-by-step process of finding the shortest path for a sample graph.		CO4	[K ₃]
	b)	Create a binary search tree for the following numbers start from an empty binary search tree.45,26,10,60,70,30,40 Delete keys 10,60 and 45 one after the other and show the trees at each stage.		CO4	[K ₄]

Please indicate knowledge level (K₁toK₆) and Course Outcome level (CO1 to CO5) against each question for each subdivision.