Practice Assignment

Module 5	
Sr. No.	**Questions**
1	Dive into the concept of Binary Trees and Binary Search Trees (BSTs). How do these data structures differ from each other? Explain the importance of the BST property in search and retrieval operations. Illustrate with a case study where using a Binary Search Tree would be advantageous.
2	Discuss the significance of height-balanced BSTs, specifically AVL Trees. What is the balancing factor, and how is it maintained? Explain the process of AVL Tree rotations and how they ensure the tree remains balanced. Provide an example of an unbalanced tree and demonstrate the rotations needed to restore balance.
3	Explore the concept of Priority Queues and their application in various scenarios. Describe the fundamental properties of a Priority Queue and how it maintains order based on priority levels. Provide a real-world example where a Priority Queue is used to solve a specific problem efficiently.
4	Delve into the world of Heaps as a specialised form of a Priority Queue. Explain the differences between Max Heaps and Min Heaps, highlighting their structures and how they maintain the heap property. Describe a practical scenario where a Max Heap would be used to address a specific task.

5	Imagine you're designing a task scheduling application that needs to handle a large number of tasks based on their priorities. Explain how you could utilise a Priority Queue (or Heap) to efficiently manage and execute tasks in the correct order. Provide step-by-step details of how the Priority Queue or Heap would be integrated into the application's architecture.
6	Suppose you are developing a spell-checking feature for a word processing software. Describe how you could employ HashTables to store a dictionary of valid words for quick and efficient word lookup. Discuss any potential challenges or trade-offs that might arise during the implementation process.
7	You are tasked with implementing a contact management system for a social networking platform. Explain how HashMaps could be utilised to store user profiles and efficiently retrieve user information using unique identifiers. Provide insights into the design considerations and hashing strategies you would apply to optimise performance and avoid collisions.
8	Consider an online shopping platform where users can add products to their cart. Describe how you could use a Binary Search Tree (BST) to maintain an ordered list of products in the cart. Discuss the benefits of using a BST over other data structures for this specific use case, and elaborate on any potential drawbacks or limitations.
9	Discuss the role of height-balanced trees, particularly AVL Trees, in optimising the performance of a search engine. Explain how AVL Trees can contribute to efficient keyword search and retrieval in a large database of documents.

	Highlight the steps involved in maintaining balance during insertion and deletion operations.
10	Imagine you are designing a simulation for an air traffic control system. Explain how you could utilise a Priority Queue (Heap) to manage incoming flight requests based on their urgency and flight paths. Discuss the specific attributes you would use for priority determination and the mechanisms to ensure timely processing of flights.