**Assignment – 11**

* **Task-1**; : Use AI to help implement a **Stack** class in Python with the following operations: push(), pop(), peek(), and is\_empty().Ask AI to generate code skeleton with docstrings.Test stack operations using sample data.Request AI to suggest optimizations or alternative implementations (e.g., using collections.deque).

**Code and output:**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Explanation:**

A screenshot of a computer program

AI-generated content may be incorrect.

* **Task-2:** Implement a **Queue** with enqueue(), dequeue(), and is\_empty() methods.First, implement using Python lists.Then, ask AI to review performance and suggest a more efficient implementation (using collections.deque).

Code and output:

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Explanation:**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

* **Task-3:** Implement a **Singly Linked List** with operations: insert\_at\_end(), delete\_value(), and traverse().Start with a simple class-based implementation (Node, LinkedList).Use AI to generate inline comments explaining pointer updates (which are non-trivial).Ask AI to suggest test cases to validate all operations.

**Code and output:**

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Explanation:**

A black screen with white text

AI-generated content may be incorrect.

**Task-4:** Implement a **Binary Search Tree** with methods for insert(), search(), and inorder\_traversal().Provide AI with a partially written Node and BST class.Ask AI to complete missing methods and add docstrings.Test with a list of integers and compare outputs of search() for present vs absent elements.

**Code and output:**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Explanation:**

A screenshot of a computer program

AI-generated content may be incorrect.

* **Task-5:** Implement a **Graph** using an adjacency list, with traversal methods BFS() and DFS().Start with an adjacency list dictionary.Ask AI to generate BFS and DFS implementations with inline comments.Compare recursive vs iterative DFS if suggested by AI.

**Code and output:**

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Explanation:**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.