Task 5

Explanation:

1. Defining the Node Class

The Node class represents a tree node:

__init__(self, value): Initializes a node with a given value and an empty list of children.

Add_child(self, child): Adds a child node to the current node.

2. Implementing DFS using a Stac

The dfs_stack(root) function performs iterative DFS:

It initializes a stack with the root node and an empty visited set.

It repeatedly pops a node from the stack, processes it, and adds its children to the stack in reversed order to maintain the correct DFS order.

3. Constructing the Tree

A (A) is the root.

B (B) and c (C) are children of A.

D (D) and e (E) are children of B.

4. Running the DFS

When dfs_stack(a) is called, it prints the nodes in Depth-First Search (DFS) order:

DFS Traversal:

A B D E C