Exercise:7

class Node:

def \_\_init\_\_(self, value):

self.value = value

self.left = None

self.right = None

class BinaryTree:

def \_\_init\_\_(self):

self.root = None

def insert(self, value):

if self.root is None:

self.root = Node(value)

else:

self.\_insert(self.root, value)

def \_insert(self, node, value):

if value < node.value:

if node.left is None:

node.left = Node(value)

else:

self.\_insert(node.left, value)

else:

if node.right is None:

node.right = Node(value)

else:

self.\_insert(node.right, value)

def inorder(self, node):

if node:

self.inorder(node.left)

print(node.value, end=' ')

self.inorder(node.right)

def preorder(self, node):

if node:

print(node.value, end=' ')

self.preorder(node.left)

self.preorder(node.right)

def postorder(self, node):

if node:

self.postorder(node.left)

self.postorder(node.right)

print(node.value, end=' ')

def display(self, node, level=0):

if node is not None:

self.display(node.right, level + 1)

print(' '\*6\*level + '/', node.value)

self.display(node.left, level + 1)

tree = BinaryTree()

while True:

print("\nBinary Tree Operations:")

print("1. Insert node")

print("2. Inorder traversal")

print("3. Preorder traversal")

print("4. Postorder traversal")

print("5. Display tree")

print("6. Exit")

choice = input("Enter your choice: ")

if choice == '1':

try:

values=input("enter node values (space-separated):").split()

for value in values:

value=int(value)

tree.insert(value)

print (f"Inserted {value} into the tree.")

except ValueError:

print("Error: please enter valid integer.")

elif choice == '2':

print("Inorder traversal: ")

tree.inorder(tree.root)

elif choice == '3':

print("Preorder traversal: ")

tree.preorder(tree.root)

elif choice == '4':

print("Postorder traversal: ")

tree.postorder(tree.root)

elif choice == '5':

print("Binary tree: ")

tree.display(tree.root)

else:

print("Invalid choice")

output:

Binary Tree Operations:

1. Insert node

2. Inorder traversal

3. Preorder traversal

4. Postorder traversal

5. Display tree

6. Exit

Enter your choice: 1

enter node values (space-separated): 20 45 32 10 55

Inserted 20 into the tree.

Inserted 45 into the tree.

Inserted 32 into the tree.

Inserted 10 into the tree.

Inserted 55 into the tree.

Binary Tree Operations:

1. Insert node

2. Inorder traversal

3. Preorder traversal

4. Postorder traversal

5. Display tree

6. Exit

Enter your choice: 2

Inorder traversal:

10 20 32 45 55

Binary Tree Operations:

1. Insert node

2. Inorder traversal

3. Preorder traversal

4. Postorder traversal

5. Display tree

6. Exit

Enter your choice: 3

Preorder traversal:

20 10 45 32 55

Binary Tree Operations:

1. Insert node

2. Inorder traversal

3. Preorder traversal

4. Postorder traversal

5. Display tree

6. Exit

Enter your choice: 4

Postorder traversal:

10 32 55 45 20

Binary Tree Operations:

1. Insert node

2. Inorder traversal

3. Preorder traversal

4. Postorder traversal

5. Display tree

6. Exit

Enter your choice: 5

Binary tree:

/ 55

/ 45

/ 32

/ 20

/ 10

Binary Tree Operations:

1. Insert node

2. Inorder traversal

3. Preorder traversal

4. Postorder traversal

5. Display tree

6. Exit

Enter your choice: 6

Invalid choice