



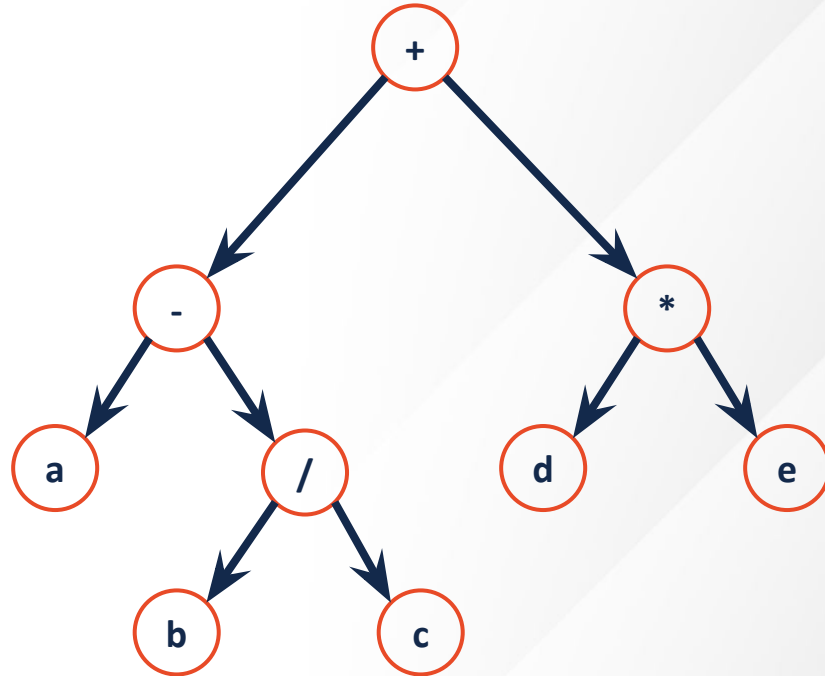
Tree Traversals

Learning Objectives

1. Implement Pre and Post Order Traversals



Accessing all nodes



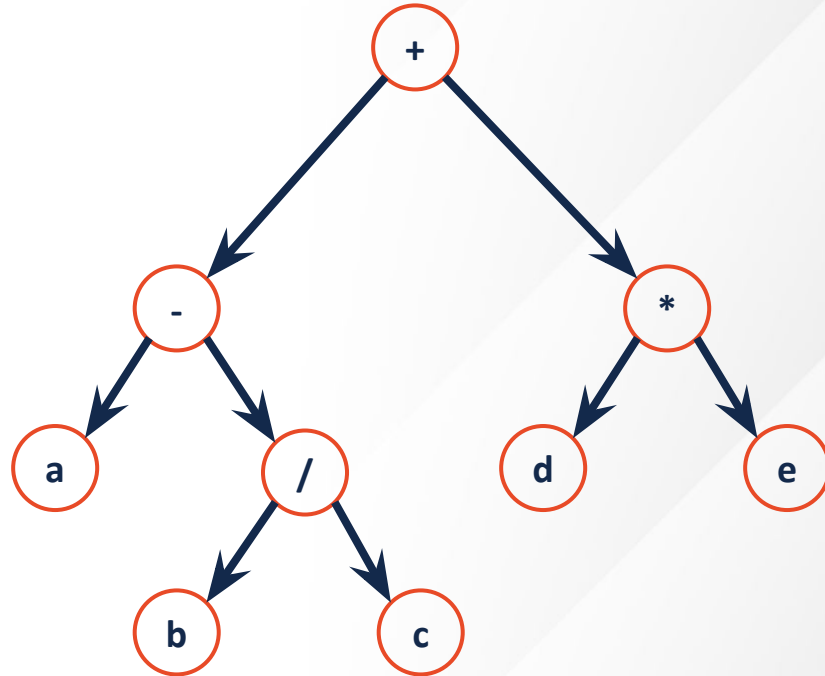
Types of Traversals

InOrder

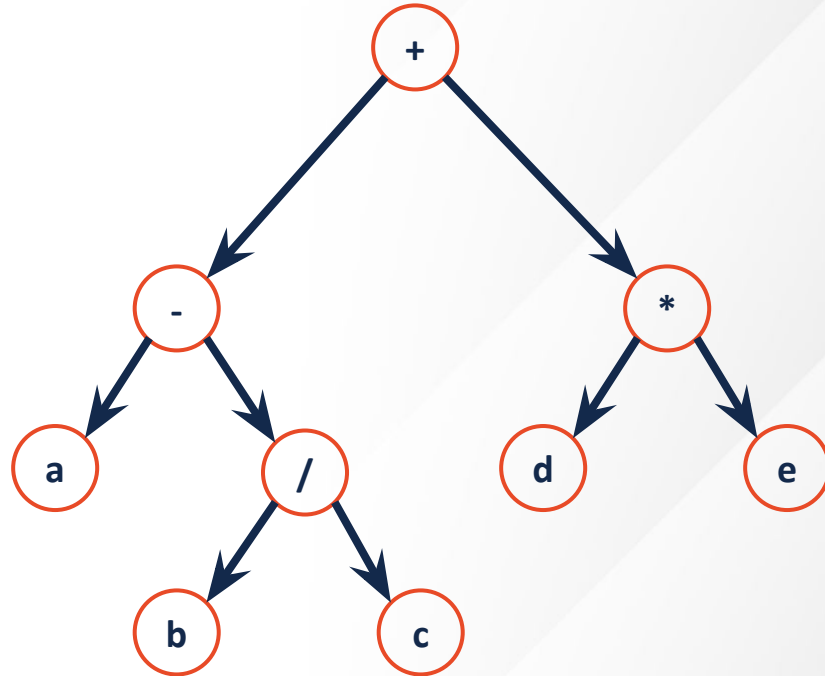
PreOrder

PostOrder

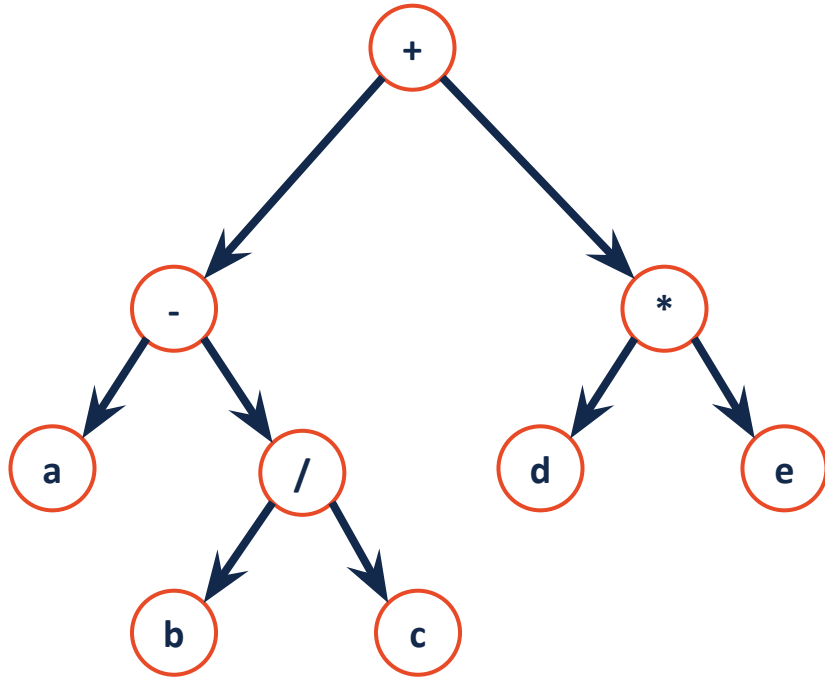
LevelOrder



Preorder Traversal

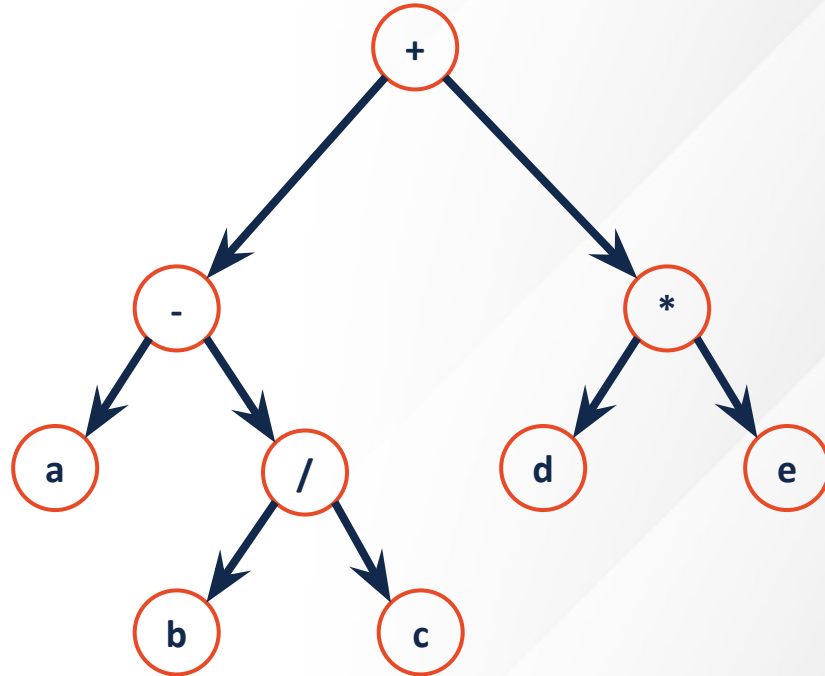


Preorder

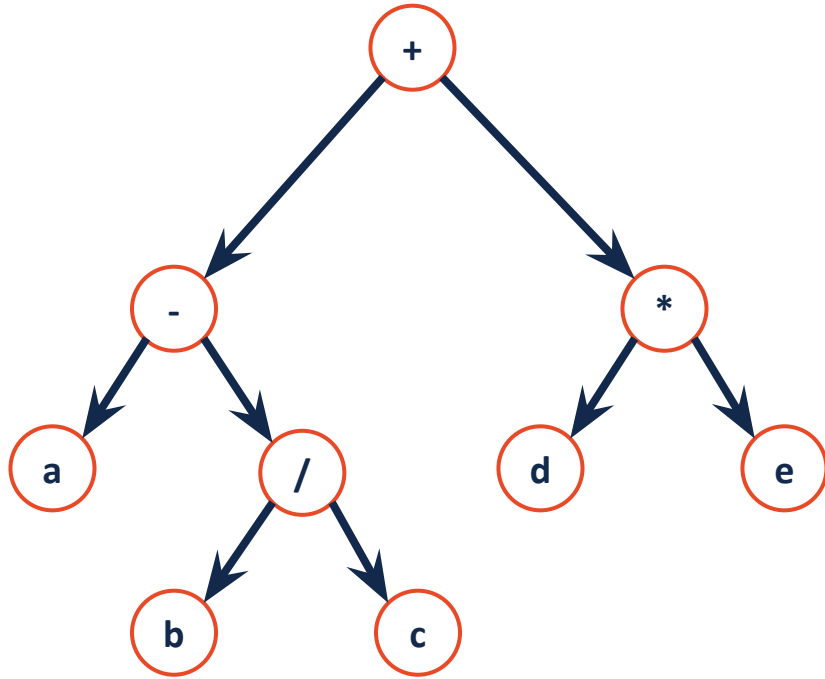


```
49 template<class T>
50 void BinaryTree<T>::PreOrder(TreeNode * cur) {
51     if (cur != NULL) {
52
53         std::cout << cur->data << std::endl;
54
55         InOrder(cur->left);
56
57         InOrder(cur->right);
58     }
59 }
```

Postorder Traversal



Postorder



```
49 template<class T>
50 void BinaryTree<T>::PostOrder(TreeNode * cur) {
51     if (cur != NULL) {
52         InOrder(cur->left);
53         InOrder(cur->right);
54         std::cout << cur->data << std::endl;
55     }
56 }
57
```

Combining information

Preorder Output: + - a / b c * d e

Postorder Output: a b c / - d e * +

