

```
#include <iostream>
using namespace std;
```

```
class Node {
public:
    int data;
    Node* left;
    Node* right;
```

```
    Node(int value) {
        data = value;
        left = right = nullptr;
    }
};
```

```
void preOrderTraversal(Node* root) {
    if (root == nullptr) return;
    cout << root->data << " ";
    preOrderTraversal(root->left);
    preOrderTraversal(root->right);
}
```

```
void inOrderTraversal(Node* root) {
    if (root == nullptr) return;
    inOrderTraversal(root->left);
    cout << root->data << " ";
    inOrderTraversal(root->right);
}
```

```
void postOrderTraversal(Node* root) {
    if (root == nullptr) return;
    postOrderTraversal(root->left);
    postOrderTraversal(root->right);
    cout << root->data << " ";
}
```

```
int main() {
    Node* root = new Node(1);
    root->left = new Node(2);
    root->right = new Node(3);
```

```
root->left->left = new Node(4);  
root->left->right = new Node(5);
```

```
cout << "Pre Order: ";  
preOrderTraversal(root);  
cout << endl;
```

```
cout << "In Order: ";  
inOrderTraversal(root);  
cout << endl;
```

```
cout << "Post Order: ";  
postOrderTraversal(root);  
cout << endl;
```

```
return 0;  
}
```

```
    1  
   / \  
  2   3  
 / \  
4   5
```

Preorder (Root-Left-Right): 1 2 4 5 3

Inorder (Left-Root-Right): 4 2 5 1 3

Postorder (Left-Right-Root): 4 5 2 3 1