

# 컴퓨터공학 All in One

---

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)  
제 30강 - 이진 트리의 구현 및 순회

# 학습 목표

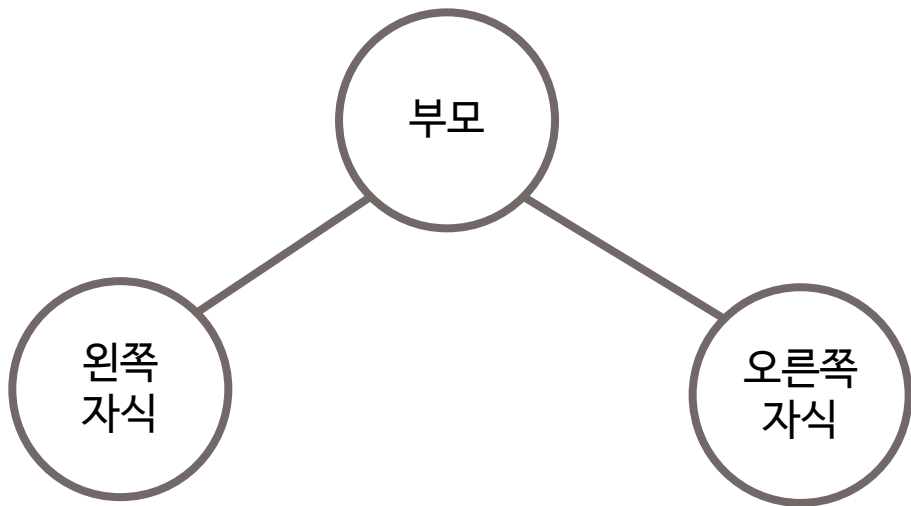
## 이진 트리의 구현 및 순회

- 1) 이진 트리의 구현 방법과 순회의 개념에 대해서 이해할 수 있습니다.
- 2) 이진 트리를 C언어로 구현하는 방법에 대해서 알 수 있습니다.

# 이진 트리의 구현 및 순회

## 이진 트리

이진 트리는 포인터를 이용하여 구현하면 효과적인 데이터 관리가 가능합니다.



# 이진 트리의 구현 및 순회

## 이진 트리

이진 트리는 포인터를 이용하여 구현하면 효과적인 데이터 관리가 가능합니다.

```
#include <stdio.h>
#include <stdlib.h>

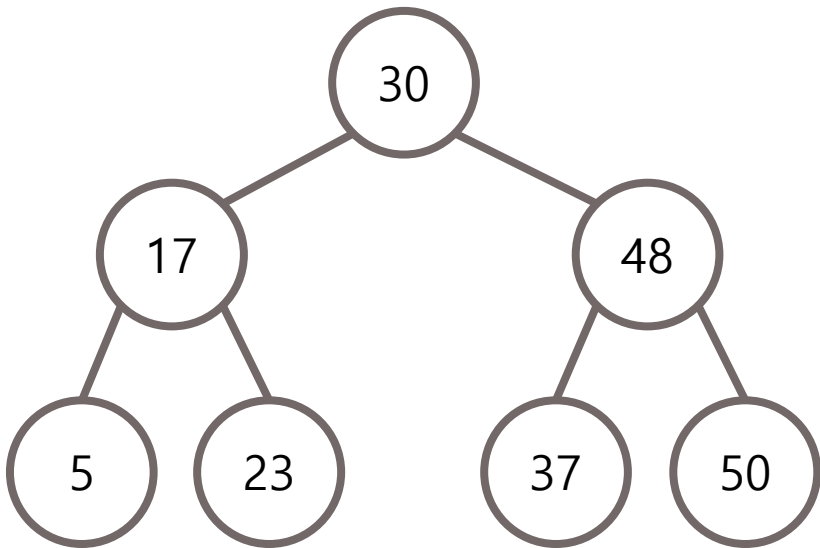
typedef struct {
    int data;
    struct Node *leftChild;
    struct Node *rightChild;
} Node;

Node* initNode(int data, Node* leftChild, Node* rightChild) {
    Node* node = (Node*)malloc(sizeof(Node));
    node->data = data;
    node->leftChild = leftChild;
    node->rightChild = rightChild;
    return node;
}
```

# 이진 트리의 구현 및 순회

## 이진 트리의 순회

이진 트리에 담긴 데이터를 하나씩 방문하는 방법으로는 대표적으로 세 가지가 있습니다.

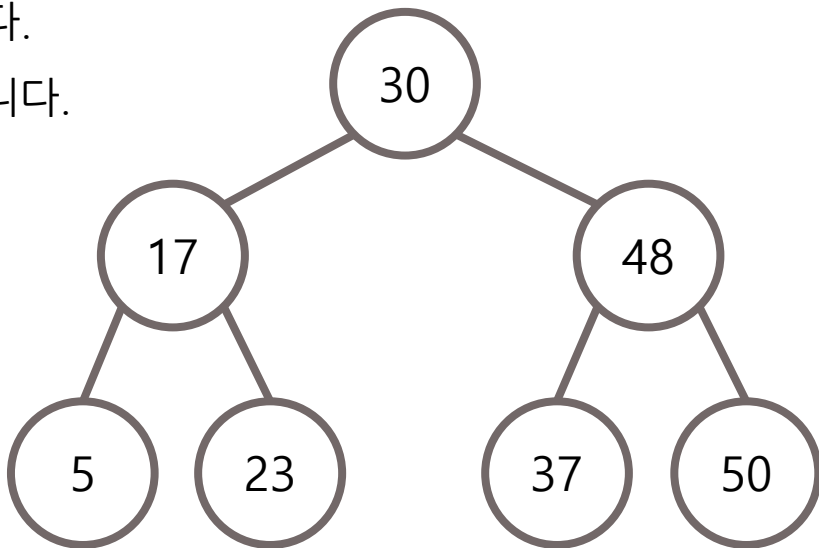


# 이진 트리의 구현 및 순회

## 이진 트리의 전위 순회

- 1) 자기 자신을 출력합니다.
- 2) 왼쪽 자식을 방문합니다.
- 3) 오른쪽 자식을 방문합니다.

출력 내용: 30 - 17 - 5 - 23 - 48 - 37 - 50



# 이진 트리의 구현 및 순회

## 이진 트리의 전위 순회

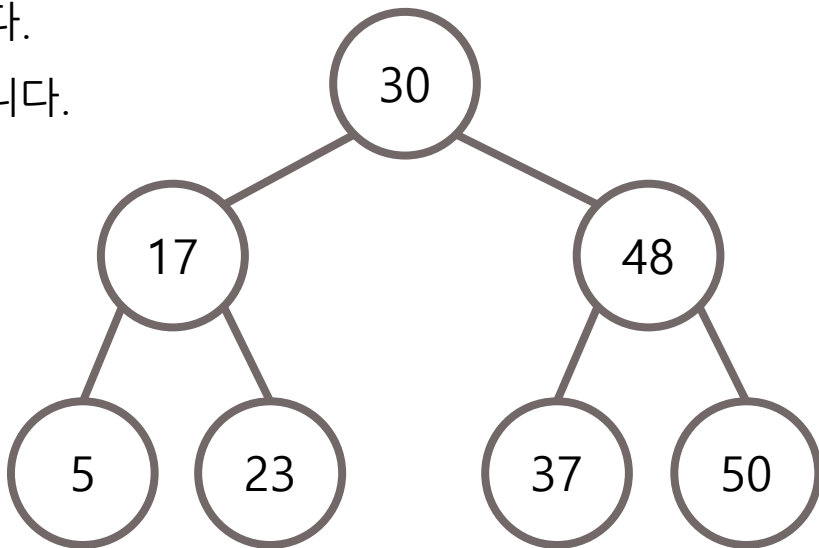
```
void preorder(Node* root) {  
    if (root) {  
        printf("%d ", root->data);  
        preorder(root->leftChild);  
        preorder(root->rightChild);  
    }  
}
```

# 이진 트리의 구현 및 순회

## 이진 트리의 중위 순회

- 1) 왼쪽 자식을 방문합니다.
- 2) 자기 자신을 출력합니다.
- 3) 오른쪽 자식을 방문합니다.

출력 내용: 5 - 17 - 23 - 30 - 37 - 48 - 50





# 이진 트리의 구현 및 순회

## 이진 트리의 중위 순회

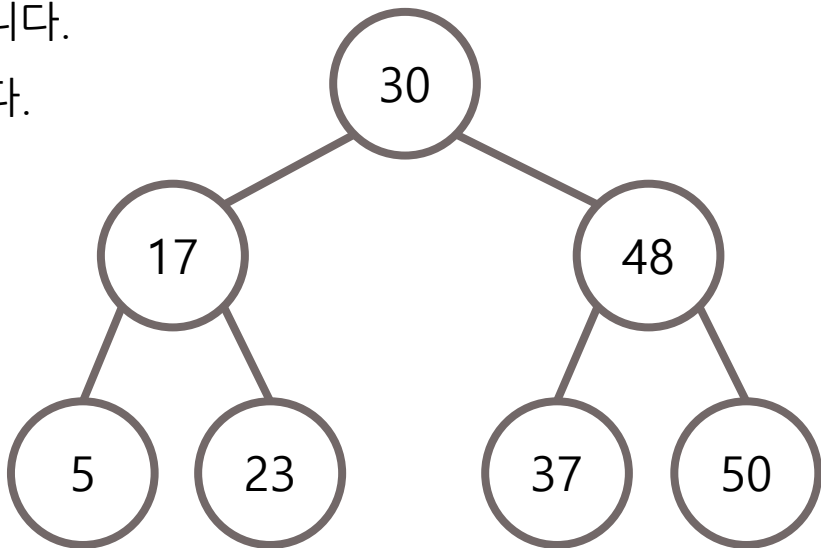
```
void inorder(Node* root) {  
    if (root) {  
        inorder(root->leftChild);  
        printf("%d ", root->data);  
        inorder(root->rightChild);  
    }  
}
```

# 이진 트리의 구현 및 순회

## 이진 트리의 후위 순회

- 1) 왼쪽 자식을 방문합니다.
- 2) 오른쪽 자식을 방문합니다.
- 3) 자기 자신을 출력합니다.

출력 내용: 5 - 23 - 17 - 37 - 50 - 48 - 30



# 이진 트리의 구현 및 순회

## 이진 트리의 후위 순회

```
void postorder(Node* root) {  
    if (root) {  
        postorder(root->leftChild);  
        postorder(root->rightChild);  
        printf("%d ", root->data);  
    }  
}
```

# 이진 트리의 구현 및 순회

## 이진 트리 사용해보기

```
int main(void) {  
    Node* n7 = initNode(50, NULL, NULL);  
    Node* n6 = initNode(37, NULL, NULL);  
    Node* n5 = initNode(23, NULL, NULL);  
    Node* n4 = initNode(5, NULL, NULL);  
    Node* n3 = initNode(48, n6, n7);  
    Node* n2 = initNode(17, n4, n5);  
    Node* n1 = initNode(30, n2, n3);  
    preorder(n1);  
    printf("\n");  
    inorder(n1);  
    printf("\n");  
    postorder(n1);  
    system("pause");  
    return 0;  
}
```

# 배운 내용 정리하기

## 이진 트리의 구현 및 순회

- 1) 이진 트리는 포인터를 이용해서 구현할 수 있습니다.
- 2) 이진 트리의 데이터를 방문하기 위해서 순회 알고리즘을 효과적으로 사용할 수 있습니다.