

정수형 데이터를 단순 연결리스트로 관리하는 CMyList 클래스를 정의해보자.

```
#include <iostream>
using namespace std;
struct Node
{
    int data;
    Node* link;
};
class CMyList
{
public:
    CMyList() {}
    ~CMyList() { // 노드 전체 삭제(메모리 해제) }
    void Insert(int num); // 새로운 노드 생성(메모리 할당) 및 link 연결
    void PrintData();
private:
    Node* pHead = nullptr;
};
```

1. 객체는 리스트의 Head 포인터를 가지고 있다. (Node* pHead)

2. 객체 선언 시 비어있는 List 객체를 생성한다.

CMyList list;

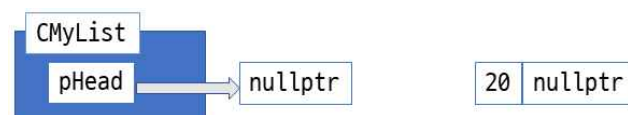
3. 데이터 삽입(Head에 데이터를 삽입한다)

list.Insert(20);

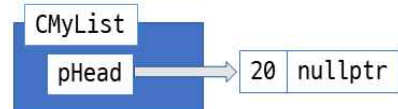
list.Insert(30);

list.Insert(40);

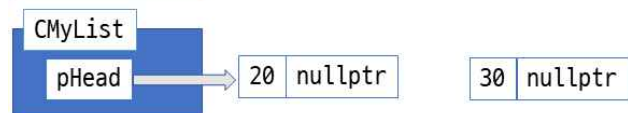
list.Insert(10);



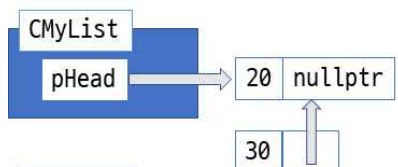
1. 사용자 입력데이터(20)로 새로운 노드 생성



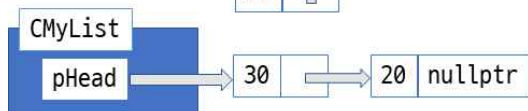
2. HeadNode 업데이트



3. 사용자 입력데이터(30)로 새로운 노드 생성



4. 새로운 노드 링크 수정



5. HeadNode 업데이트

4. 데이터 출력 (리스트에 저장된 데이터를 Head부터 차례대로 보여준다)

list.PrintData();

```
#include <iostream>
using namespace std;
int main()
{
    CMyList list;
    list.Insert(20);
    list.Insert(30);
    list.Insert(40);
    list.Insert(10);
    list.PrintData();
    return 0;
}
```

C:\Windows\system32\cmd.exe

리스트 데이터 : 10 - 40 - 30 - 20 [4개]
계속하려면 아무 키나 누르십시오 . . .