# C Programming Language

(9th class)

**Dohyung Kim**

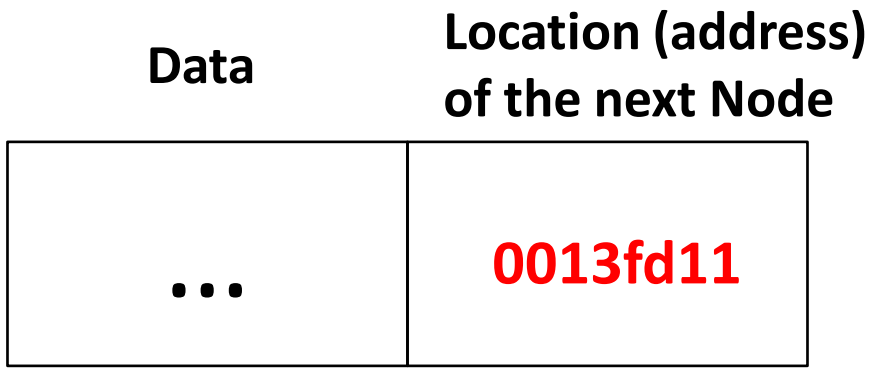
Assistant Professor @ Department of Computer Science

# Today …

- Review Linked List Data Structure
- Programming with Linked List Data Structure

# Linked List

- A linked list is composed of nodes
- Node



# Linked List in C

- node structure that comprises a linked list

```
struct node {
   int val;
   struct node *nextNode;
};
```

data

pointer (address, arrow)

- Creation of an empty linked list

```
struct node *myLinkedList = NULL;
```

myLinkedList

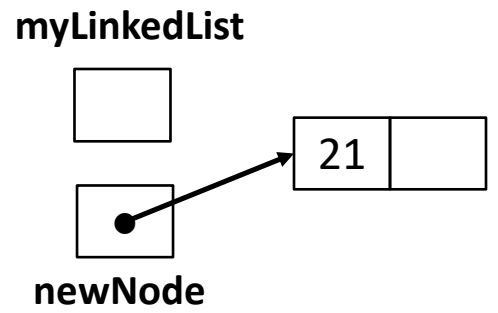
# Linked List in C

- **Create a new node for a new data**

```
struct node *newNode = (struct node*) malloc ( sizeof ( struct node));

newNode->val = 21;
newNode->nextNode = NULL;
```

create a memory space for a new node

data and pointer values are set



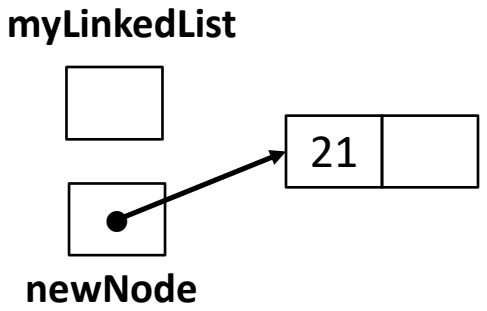
# Linked List in C

- **Create a new node for a new data**

```
struct node *newNode = (struct node*) malloc ( sizeof ( struct node));

newNode->val = 21;
newNode->nextNode = NULL;
```

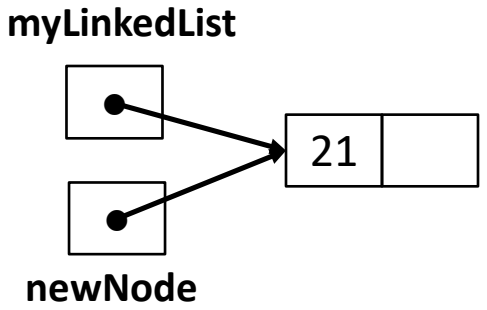
create a memory space for a new node

data and pointer values are set




- **Add the first new node to the link**

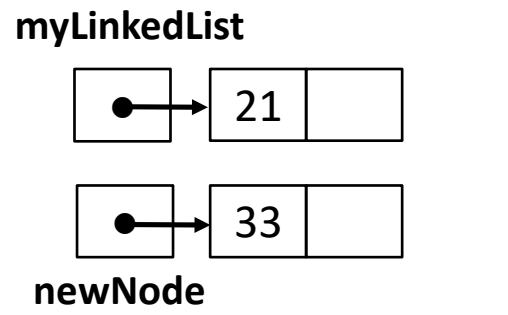
```
myLinkedList = newNode;
```



# Linked List in C

- **Create another new node for the value of 33**

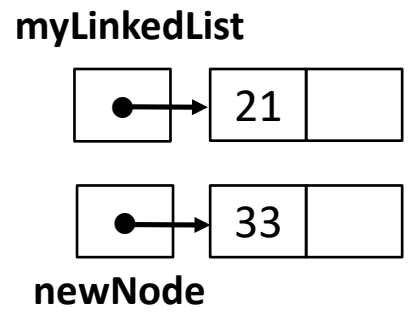
```
struct node *newNode = (struct node*) malloc ( sizeof ( struct node));

newNode->val = 33;
newNode->nextNode = NULL;
```



# Linked List in C

- **Create another new node for the value of 33**

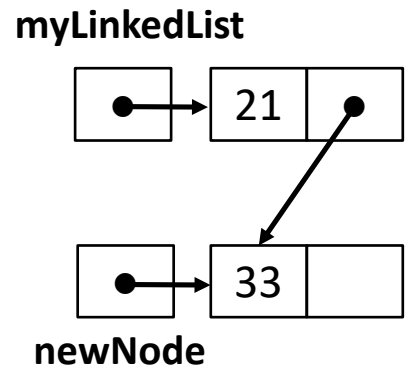
```
struct node *newNode = (struct node*) malloc ( sizeof ( struct node));

newNode->val = 33;
newNode->nextNode = NULL;
```




- **Add the new node to the end of the list**

```
struct node *curPos = NULL;
curPos = myLinkedList;
while (curPos->next != NULL)
   curPos = curPos->next;
curPos->next = newNode;
```

**curPos points to the end node in the list, and the new node connects behind that end node.**



# Today’s program

1. Add ten numbers to a linked list (A)
   - 10개의 숫자를 linked list (A) 에 넣을수 있도록 합니다.
2. Create a new linked list (B) by finding a number of two in the linked list (A)
   - (A)에서 2의 배수들을 찾아서 새로운 linked list를 만듭니다.
3. Practice in the Lab Session :  Remove numbers in (B) from (A)
   - (B)에 속하는 숫자들을 linked list (A) 에서 삭제하도록 합니다.

# Skeleton Code

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

struct node {
…
};

void printList(struct node* list) {
…
}

struct node* insertNodeToList (struct node* list, int val){
…
}

struct node * listMultipleOfTwo (struct node* list){
…
}
```

```
void mainJ() {
   struct node *myLinkedList = NULL;
   int inVal;

   for (int i=0; i<10; i++){
      printf("\n input your number: ");
      scanf("%d", &inVal);
      myLinkedList = insertNodeToList ( myLinkedList, inVal );
   }


   printList ( myLinkedList );
   printList ( listMultipleofTwo ( myLinkedList ) );
```

# Q and A