

# C++ STANDARD TEMPLATE LIBRARY (STL) (LIST)

**SETHALAT RODHETBHAI**

DEPARTMENT OF COMPUTING • SILPAKORN UNIVERSITY

03/2023

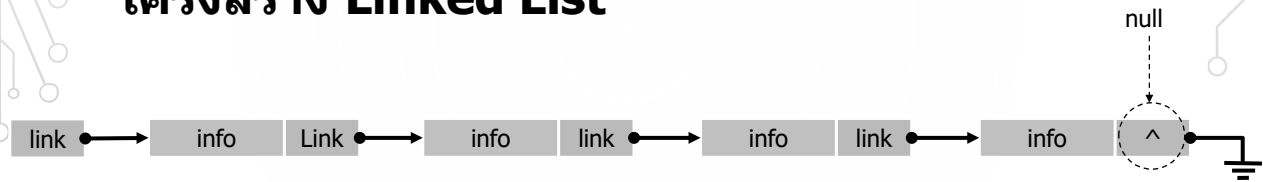
## Linked Lists

- Linked Lists
  - List ของ item, เรียกว่า **node**
  - ลำดับของ node ถูกจัดการโดยใช้ตำแหน่ง address, เรียกว่า **link**, ซึ่งเก็บอยู่ใน node
- แต่ละ node (*ยกเว้น node สุดท้าย*) ประกอบไปด้วย ตำแหน่ง address (link) ของ node ถัดไป
- องค์ประกอบของ node
  - **Info (Data)**: เก็บ information
  - **Link**: เก็บตำแหน่ง address ของ node ถัดไป

info

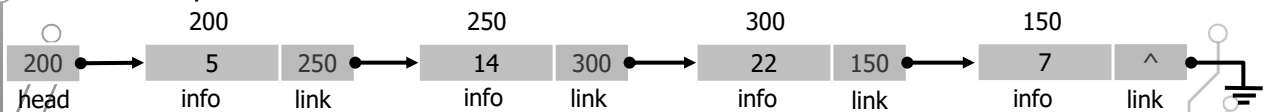
link

## โครงสร้าง Linked List



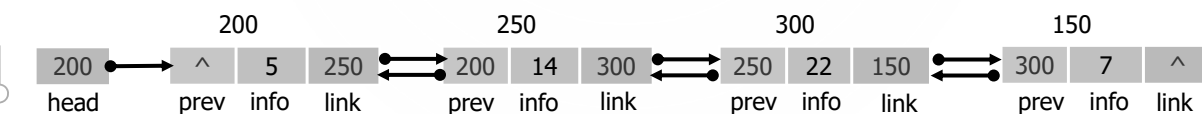
- **head** หรือ **first** ใช้เก็บ address ของ node แรก ใน list
- **info** ใช้เก็บข้อมูลใน node สามารถที่จะเป็น primitive type หรือ reference ไปยัง object ก็ได้

### Example



## List สองทิศทาง (Doubly Linked List)

- ในแต่ละ node จะมีการอ้างอิงตำแหน่ง address อยู่ 2 ค่า ได้แก่
  - ค่า **link** อ้างอิงไปยังตำแหน่ง address ของ node ที่อยู่ถัดไป
  - ค่า **prev** อ้างอิงไปยังตำแหน่ง address ของ node ที่อยู่ก่อนหน้า



# การสร้าง List Container

## รูปแบบ

```
list<data_type> list_name;
```

## ตัวอย่าง

```
list<double> scores;  
list<string> names;  
list<int> ages;
```

# การสร้าง List Container ของ Record (Struct)

## ตัวอย่าง

```
struct Student {  
    int no;  
    string name;  
    double score;  
};  
list<Student> classroom;
```

## การสร้าง List Container ของ Object

### ตัวอย่าง

```
class Person {  
    public:  
        string name;  
        int age;  
        double income;  
}  
  
list<Person> people;
```

## การสร้าง List Container พร้อมค่าเริ่มต้น

### ตัวอย่าง

```
list<int> month{31,28,31,30,31,30,31,31,30,31,30,31};  
for(int t : month)  
    cout << t << " ";  
  
list<double> height(10,0);  
for(double t : height)  
    cout << t << " ";
```

## การสร้าง List Container จากข้อมูลอื่น

### ตัวอย่าง

```
int month_arr[]={31,28,31,30,31,30,31,31,30,31,30,31};  
int n=sizeof(month_arr)/sizeof(month_arr[0]);  
list<int> month(month_arr,month_arr+n);  
for(int t : month)  
    cout << t << " ";
```

## การสร้าง List Container จากข้อมูลอื่น

### ตัวอย่าง

```
vector<int> month_vec{31,28,31,30,31,30,31,31,30,31,30,31};  
list<int> month(month_vec.begin(),month_vec.end());  
for(int t : month)  
    cout << t << " ";
```

```
list<int> month_lst{31,28,31,30,31,30,31,31,30,31,30,31};  
list<int> month(month_lst.begin(),month_lst.end());  
for(int t : month)  
    cout << t << " ";
```

## การสร้าง List Container และใส่ค่าเป็นชุด

### ตัวอย่าง

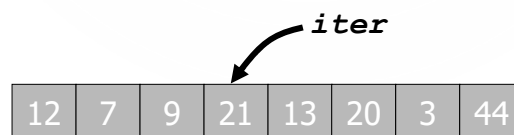
```
list<float> height(10);  
fill(height.begin(), height.end(), 100);  
for(float t : height)  
    cout << t << " ";
```

```
list<double> height;  
height.assign(15, 100);  
for(double t : height)  
    cout << t << " ";
```

## Iterator ของ List Container

### ตัวอย่าง

```
list<int> lst{12, 7, 9, 21, 13, 20, 3, 44};  
list<int>::iterator iter;  
for(iter=lst.begin(); iter!=lst.end(); iter++)  
    cout << (*iter) << " ";
```



## Method ของ List Container

- เมธอดที่สำคัญของ List Container เช่น

<b>push_back();</b>	// เพิ่มข้อมูลต่อในตำแหน่งสุดท้ายของ list
<b>push_front();</b>	// เพิ่มข้อมูลต่อในตำแหน่งหน้าสุดของ list
<b>pop_back()</b>	// ลบข้อมูลในตำแหน่งท้ายสุดของ list
<b>pop_front()</b>	// ลบข้อมูลในตำแหน่งหน้าสุดของ list
<b>front();</b>	// คืนค่าข้อมูลในตำแหน่งแรกสุดของ list
<b>back();</b>	// คืนค่าข้อมูลในตำแหน่งท้ายสุดของ list
<b>size();</b>	// คืนค่าจำนวนข้อมูลใน list
<b>empty();</b>	// ตรวจสอบว่า list ว่างหรือไม่
<b>remove();</b>	// ลบค่าข้อมูลใน list
<b>insert();</b>	// แทรกข้อมูลใน list ณ ตำแหน่ง iterator
<b>erase();</b>	// ลบข้อมูลใน list ณ ตำแหน่ง iterator
<b>unique();</b>	// ทำให้ข้อมูลใน list ไม่ซ้ำกัน
<b>reverse();</b>	// เรียงข้อมูลใน list ในลำดับย้อนกลับ
<b>merge();</b>	// รวมข้อมูลของ list อื่นเข้ากับข้อมูลใน list
<b>sort();</b>	// จัดเรียงลำดับข้อมูลใน list

## การเพิ่มและลบข้อมูลใน List Container

```
int array[8] = {12,7,9,21,13,20,3,44};
```

```
list<int> lst(array,array+5);
```

12	7	9	21	13
----	---	---	----	----

```
lst.pop_back();
```

12	7	9	21
----	---	---	----

13

```
lst.push_back(15);
```

12	7	9	21	15
----	---	---	----	----

...

```
lst.pop_front();
```

12

7	9	21
---	---	----

...

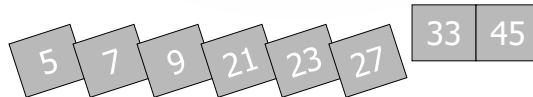
```
lst.push_front(8);
```

8	12	7	9	21	15
---	----	---	---	----	----

## ข้อมูลใน List Container ณ ตำแหน่งแรก

### ตัวอย่าง

```
list<int> lst{5,7,9,21,23,27,33,45};  
  
int val=30;  
while(lst.size()>0 && lst.front()<val) {  
    cout << lst.front() << " ";  
    lst.pop_front();  
}
```



## ข้อมูลใน List Container ณ ตำแหน่งสุดท้าย

### ตัวอย่าง

```
list<int> lst{5,7,9,21,23,27,33,45};  
  
int val=30;  
while(!lst.empty() && lst.back()>val) {  
    cout << lst.back() << " ";  
    lst.pop_back();  
}
```

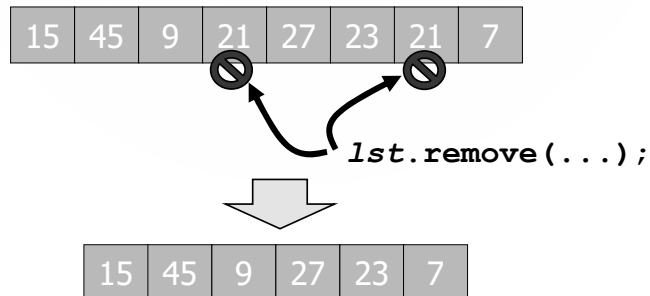




## การลบค่าข้อมูลใน List Container

### ตัวอย่าง

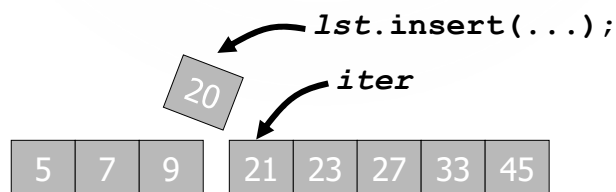
```
list<int> lst{15,45,9,21,27,23,21,7};  
int val=21;  
lst.remove(val);
```



## การแทรกข้อมูลใน List Container ณ ตำแหน่ง Iterator

### ตัวอย่าง

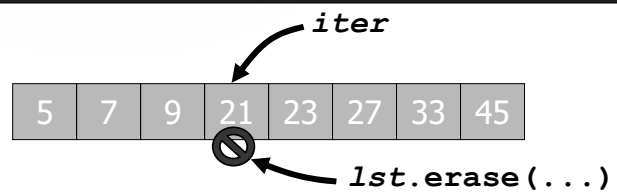
```
list<int> lst{5,7,9,21,23,27,33,45};  
int val=20;  
  
list<int>::iterator iter;  
for(iter=lst.begin(); (*iter) <= val; iter++) ; //null statement  
lst.insert(iter,val);
```



## การลบข้อมูลใน List Container ณ ตำแหน่ง Iterator

### ตัวอย่าง

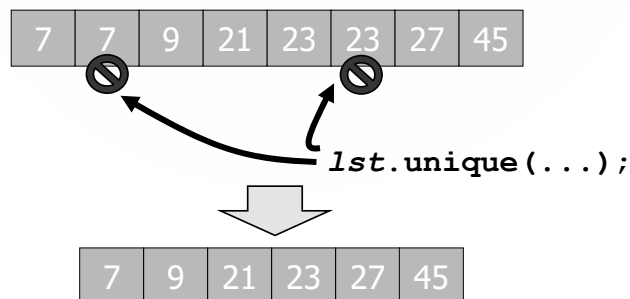
```
list<int> lst{5,7,9,21,23,27,33,45};  
int val=21;  
  
list<int>::iterator iter;  
for(iter=lst.begin(); (*iter)!=val&&iter!=lst.end(); iter++)  
    ; //null statement  
if(iter!=lst.end())  
    lst.erase(iter);
```



## การกำจัดค่าข้อมูลใน List Container ที่ซ้ำกัน

### ตัวอย่าง

```
list<int> lst{7,7,9,21,23,23,27,33}; //sorted  
lst.unique();
```

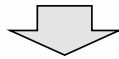


## การสลับตำแหน่งข้อมูลใน List Container จากหลังไปหน้า

### ตัวอย่าง

```
list<int> lst{9,21,13,27,21,9,45,24};  
lst.reverse();
```

9	21	13	27	21	9	45	24
---	----	----	----	----	---	----	----



24	45	9	21	27	13	21	9
----	----	---	----	----	----	----	---

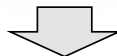
## การผนวก List Container เข้าด้วยกัน

### ตัวอย่าง

```
list<int> lst1{5,7,9,21,23,27,33,45};           //sorted  
list<int> lst2{1,2,19,25,29,30,44,49};         //sorted  
lst1.merge(lst2);
```

lst1	5	7	9	21	23	27	33	45
------	---	---	---	----	----	----	----	----

lst2	1	2	19	25	29	30	44	49
------	---	---	----	----	----	----	----	----



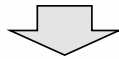
lst1	1	2	5	7	9	19	21	23	25	27	29	30	33	44	45	49
------	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

## การเรียงลำดับค่าข้อมูลใน List Container

### ตัวอย่าง

```
list<int> lst{9,21,13,27,21,9,45,24};  
lst.sort();
```

9	21	13	27	21	9	45	24
---	----	----	----	----	---	----	----



9	9	13	21	21	24	27	45
---	---	----	----	----	----	----	----