



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

**WASARA RODHETBHAJ**

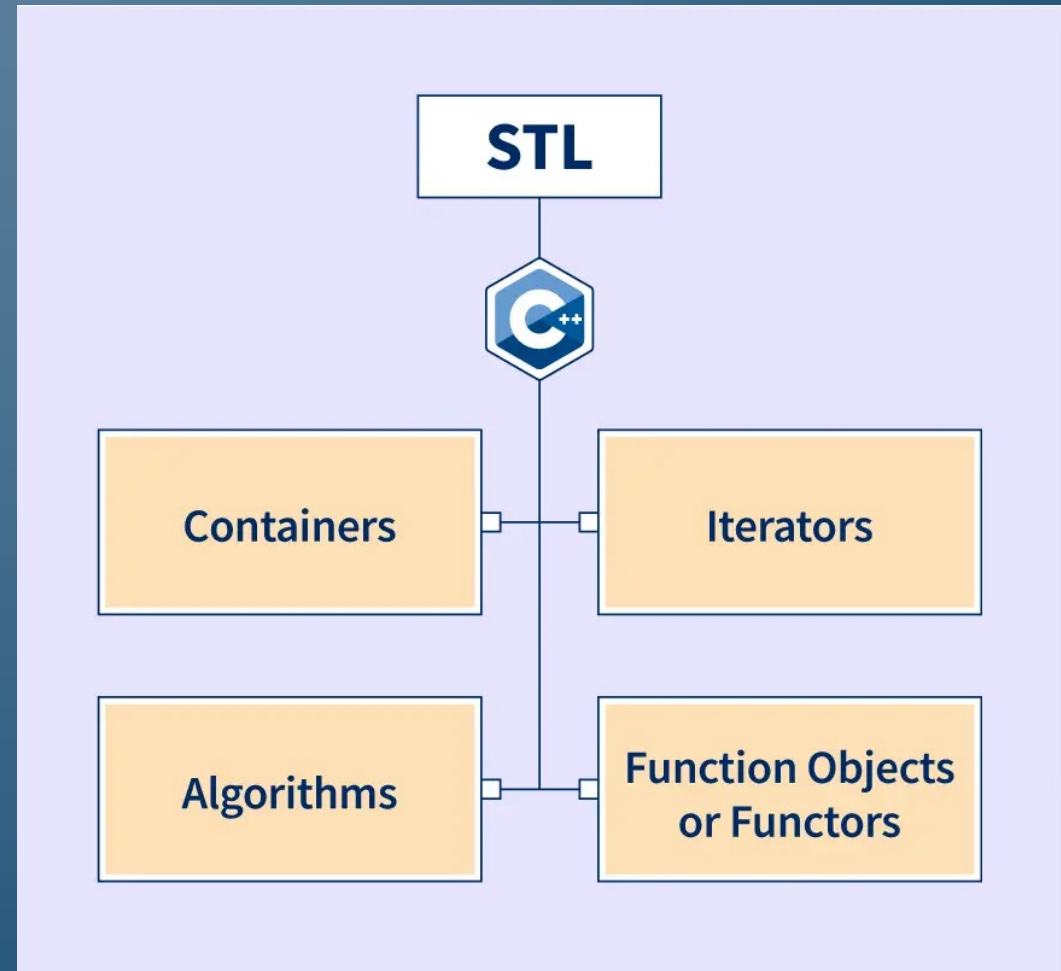
---

**DEPARTMENT OF COMPUTING • SILPAKORN UNIVERSITY**

# Standard Template Library (STL)

## 4 ส่วนประกอบหลักของ STL

- Containers
- Iterators
- Algorithms
- Function objects



# Containers

```
graph TD; Containers[Containers] --> Sequence[Sequence Containers]; Containers --> Associative[Associative Containers]; Containers --> Derived[Derived Containers (Container Adaptors)];
```

## Sequence Containers

- Vector
- Deque
- List

## Associative Containers

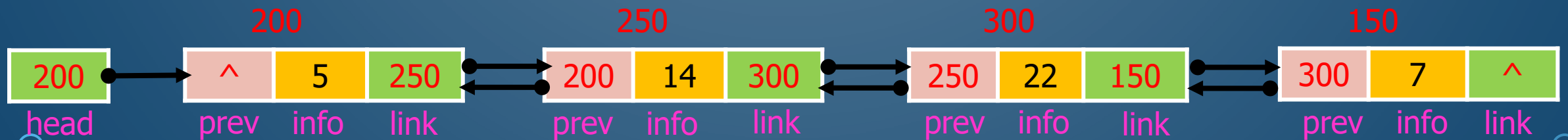
- Set
- Multiset
- Map
- Multimap

## Derived Containers (Container Adaptors)

- Stack
- Queue
- Priority\_Queue

# List Container

โครงสร้างข้อมูลเชิงเส้นที่มีการจัดเก็บข้อมูลที่สามารถเพิ่มและลบข้อมูลได้ทั้งสองทิศทาง โดย List ไม่จำเป็นต้องเก็บข้อมูลไว้ในหน่วยความจำที่ติดกัน ซึ่งต่างกับ Vector



Doubly-linked list Data Structure

# การสร้าง List Container

```
#include <list>
```

## 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

**push\_back();**

// เพิ่มข้อมูลต่อในตำแหน่งท้ายสุดของ list

**push\_front();**

// เพิ่มข้อมูลต่อในตำแหน่งหน้าสุดของ list

**pop\_back();**

// ลบข้อมูลในตำแหน่งท้ายสุดของ list

**pop\_front ();**

// ลบข้อมูลในตำแหน่งหน้าสุดของ list

**front();**

// คืนค่าข้อมูลในตำแหน่งแรกสุดของ list

**back();**

// คืนค่าข้อมูลในตำแหน่งท้ายสุดของ list

**size();**

// คืนค่าจำนวนข้อมูลใน list

**empty();**

// ตรวจสอบว่า list ว่างหรือไม่

**remove();**

// ลบค่าข้อมูลใน list

**insert();**

// แทรกข้อมูลใน list ณ ตำแหน่ง iterator

**erase();**

// ลบข้อมูลใน list ณ ตำแหน่ง iterator

**unique();**

// ทำให้ข้อมูลใน list ไม่ซ้ำกัน

**reverse();**

// เรียงข้อมูลใน list ในลำดับย้อนกลับ

**merge();**

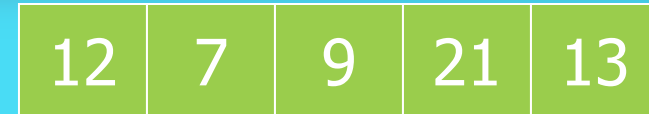
// รวมข้อมูลของ list อื่นเข้ากับข้อมูลใน list

**sort();**

// จัดเรียงลำดับข้อมูลใน list

# การเพิ่มและลบข้อมูลใน List Container

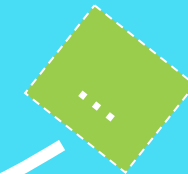
```
int array[8] = {12,7,9,21,13,20,3,44};  
list <int> lst(array,array+5);
```



*lst*.pop\_back();



*lst*.push\_back(15);



*lst*.pop\_front();



*lst*.push\_front(8);



# ข้อมูลใน List Container ณ ตำแหน่งแรก

## ตัวอย่าง

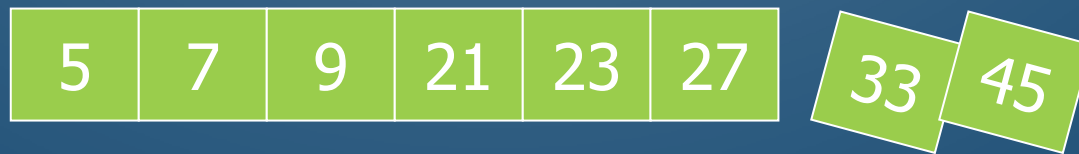
```
list<int> lst{5,7,9,21,23,27,33,45};  
int val=30;  
while(lst.size()>0 && lst.front()<val) {  
    cout << lst.front() << " ";  
    lst.pop_front();  
}
```



# 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();  
}
```





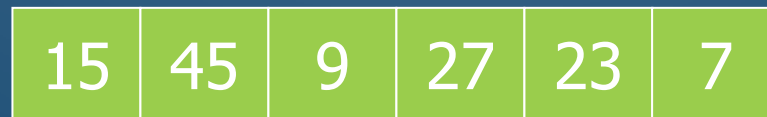
# การลบค่าข้อมูลใน List Container

## ตัวอย่าง

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



*lst.remove(...);*



# การแทรกข้อมูลใน List Container ณ ตำแหน่ง Iterator

## ตัวอย่าง

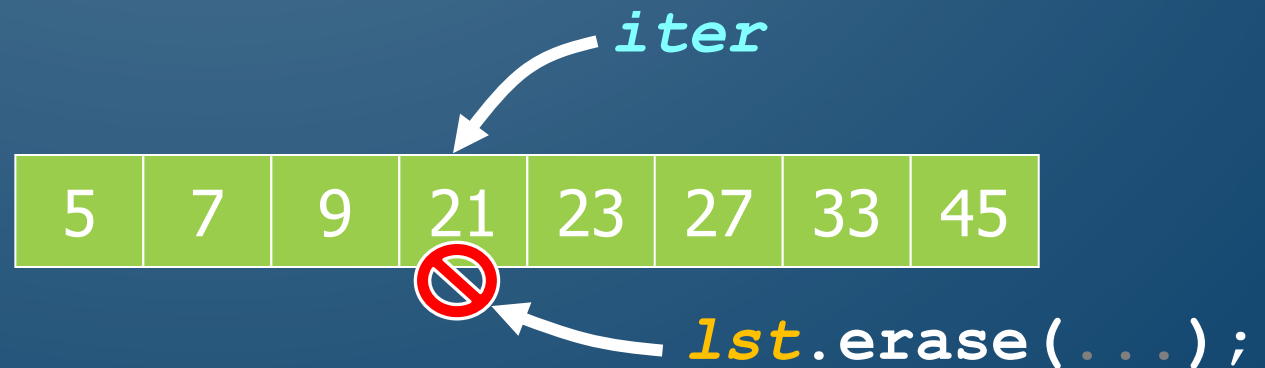
```
list<int> lst{5,7,9,21,23,27,33,45};  
int val=20;  
  
list<int>::iterator iter;  
for(iter=lst.begin(); (*iter) <= val; iter++) ; //null statement  
lst.insert(iter, val);
```



# การลบข้อมูลใน List Container ณ ตำแหน่ง Iterator

## ตัวอย่าง

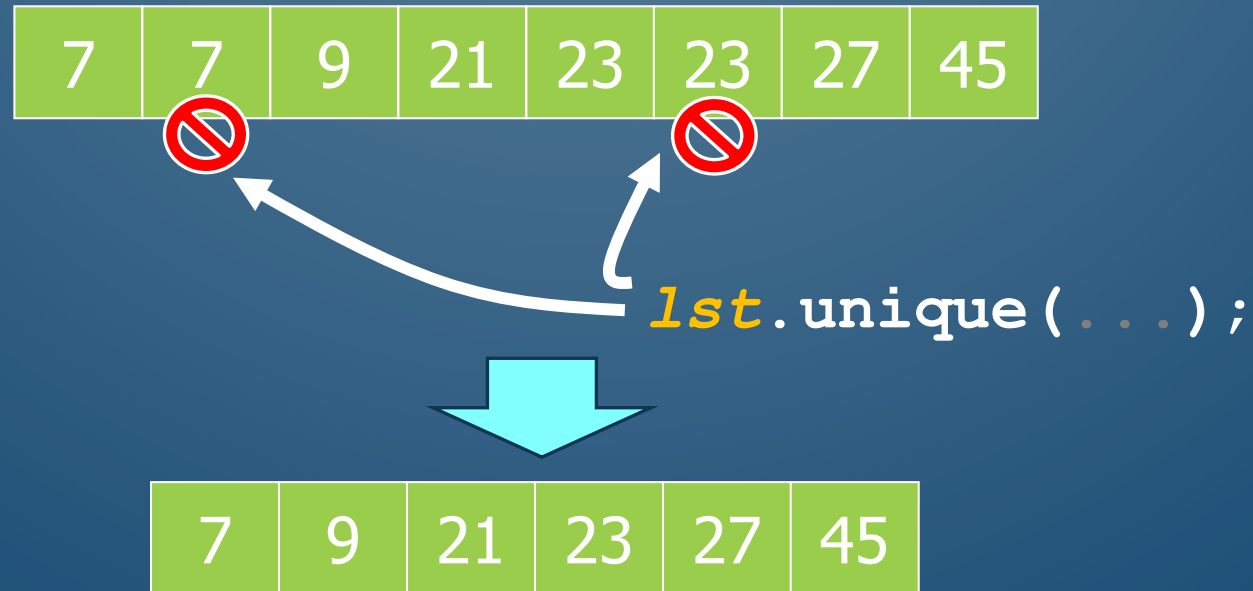
```
list<int> lst{5,7,9,21,23,27,33,45};  
int val=21;  
  
list<int>::iterator iter;  
for(iter=lst.begin(); (*iter) != val && iter != lst.end(); iter++)  
    ; //null statement  
if(iter != lst.end())  
    lst.erase(iter);
```



# 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);
```

<i>lst1</i>	5	7	9	21	23	27	33	45
<i>lst2</i>	1	2	19	25	29	30	44	49



<i>lst1</i>	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
---	---	----	----	----	----	----	----