CST209

Object-oriented Programming C++

(Week 3)

Dr. Teo Bee Guan

Content

Standard Template Library (STL)

STL - Overview

- C++ Standard Template Library (STL) is a powerful set of C++ template classes to provide general-purpose classes and functions with templates that implement many popular and commonly used algorithms and data structures

- Divided into three components:
  - Containers: data structures that store objects of any type
  - Iterators: used to manipulate container elements
  - Algorithms: searching, sorting and many others

Sequence Containers: Vector

- The implementation of a vector is based on arrays
- Vectors are used to store elements of similar data types. However, unlike arrays, the size of a vector can grow dynamically.
- We can change the size of the vector during the execution of a program as per our requirements.

STL - Overview

- There are two types of container classes in the STL: **sequence** and **associative** .
- A sequence container organizes data in a sequential fashion similar to an array. For example:
  i. Vector
  ii. Stack
  iii. Deque
  iv. Queue

STL - Overview

- On another hand, an associative container uses keys to rapidly access elements. For example:
  i. Set
  ii. Multiset
  iii. Map
  iv. multimap

Sequence Containers: Vector

- To use vectors, we need to include the vector header file in our program.

```
#include <vector>
```

- The type parameter <T> specifies the type of the vector. It can be any primitive data type such as int, char, float, etc.

Practice: Code Example 1

Sequence Containers: Vector

- The vector class provides various methods to perform different operations on vectors.
  - Add elements
  - Access elements
  - Change elements
  - Remove elements

Practice: Code Example 2

Sequence Containers (1): Vector

- There are many algorithms in the Vector, implemented as function templates. For example:
  - random_shuffle
  - sort
  - count
  - max_element
  - min_element

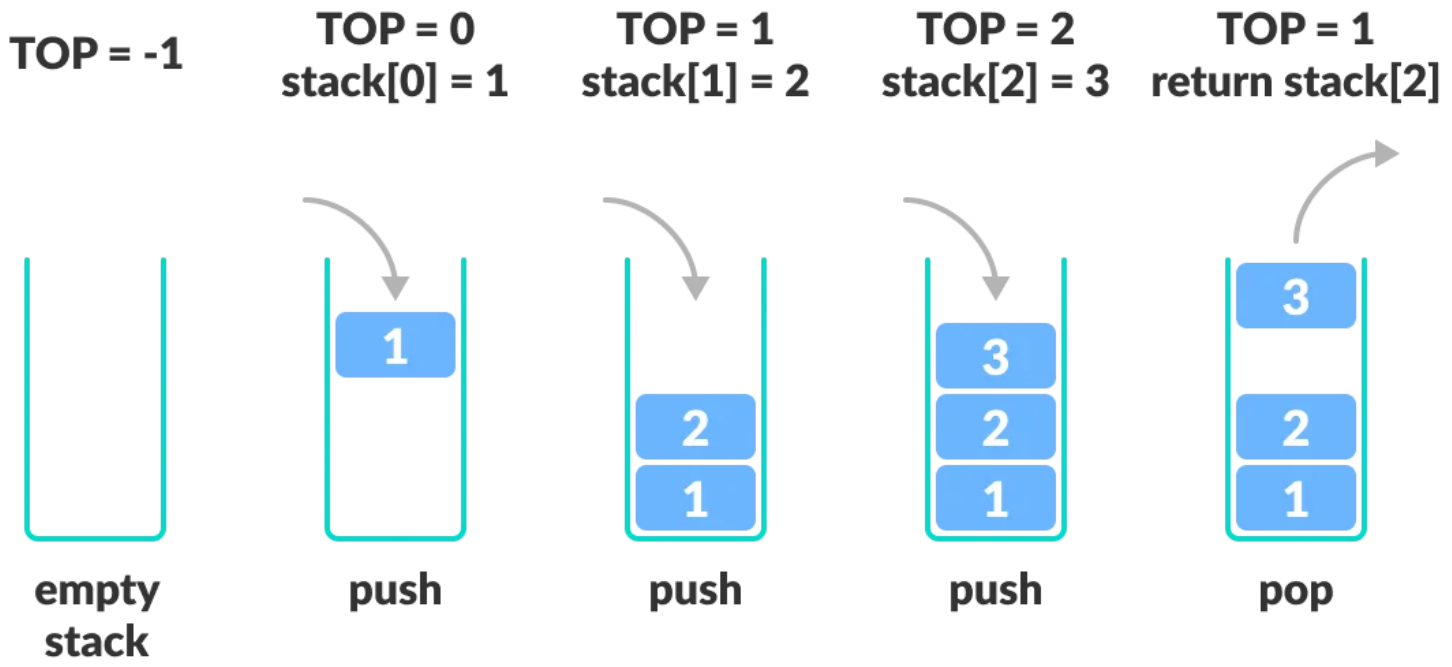
Practice: Code Example 3, 4

In-class Exercise

- Write a C++ program that lets the user enter the total rainfall for each of 12 months into a vector of doubles. The program should calculate and display the rainfall for every month and also the highest and lowest rainfall amounts.

Sequence Containers (2): Stack

- Last-in-first-out data structure



Sequence Containers (2): Stack

- They are implemented with vector, list, and deque (by default)
- Header file <stack>
- Example of creating stacks
  - A stack of int using a vector: **stack < int, vector < int > > s1;**
  - A stack of int using a list: **stack < int, list < int > > s2;**
  - A stack of int using a deque: **stack < int > s3;**

Practice: Code Example 5

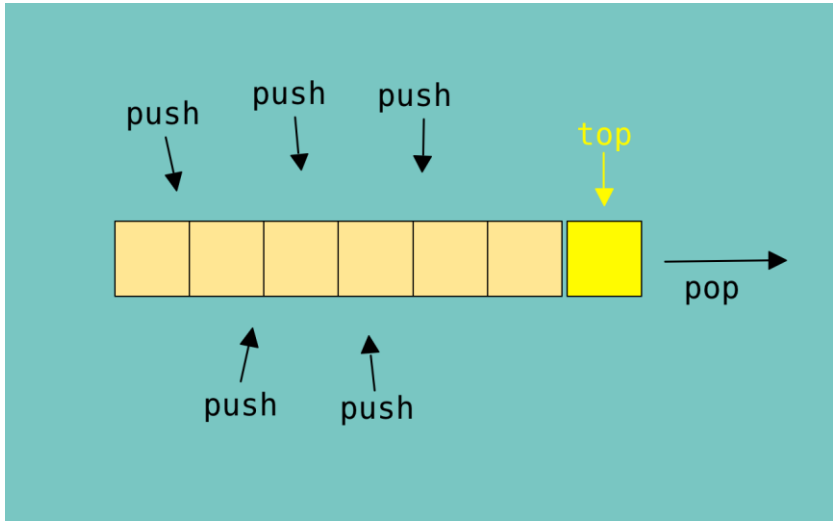
Sequence Containers (3): Deque

- deque stands for double-ended queue
- It provides indexed access using indexes
- It also provides efficient insertion and deletion in the front (which is not efficient using vectors) and the end
- Additional storage for a deque is allocated using blocks of memory that are maintained as an array of pointers to those blocks
- Same basic functions as vector, in addition to that deque supports push_front and pop_front for insertion and deletion at beginning of deque

Practice: Code Example 6

Sequence Containers (4): Queue

- First-in-first-out data structure



- Header file <queue>

- Example:
  - A queue of int using a list: **queue <int, list<int>> q1;**
  - A queue of int using a deque: **queue <int> q2;**

Practice: Code Example 7

Associative Containers

- Associative containers use keys to store and retrieve elements

- There are four types: multiset, set, multimap and map
  - **all** associative containers maintain keys in sorted order
  - **all** associative containers support bidirectional iterators
  - **set** does not allow duplicate keys
  - **multiset** and **multimap** allow duplicate keys
  - **multimap** and **map** allow keys and values to be mapped

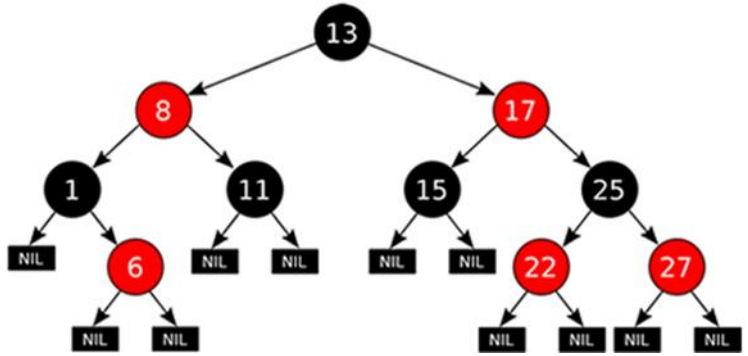
Associative Containers (1): Set

- Sets in C++ are the type of STL containers that are used for storing elements in a sorted way.
- Whenever there is a need of storing the elements in a sorted manner, we can think of using the sets, but remember that the set does not allow us to store the duplicates value.
- The value can’t be modified once it is stored.

Practice: Code Example 8

Associative Containers (2): Multiset

- Multisets are implemented using a red-black binary search tree for fast storage and retrieval of keys



- Multisets allow duplicate keys
- The ordering of the keys is determined by the STL comparator function object less<T>
- Keys sorted with less<T> must support comparison using the < operator

Practice: Code Example 9

Associative Containers (3): Map

- Maps are the associative containers that store sorted key-value pair, in which each key is unique and it can be inserted or deleted but cannot be altered.

- Values associated with keys can be changed.

Practice: Code Example 10

Associative Containers (4): Multimap

- Multimaps associate keys to values
- Insertion is done using objects of the class pair (with a key and value)
- Multimaps allow duplicate keys (many values can map to a single key)
- The ordering of the keys is determined by the STL comparator function object less<T>

Practice: Code Example 11

See you next class