Standard Template Library in C++

By – Mohammad Imran

Standard Template Library

Introduction

The **Standard Template Library or STL** in C++ is a collection of template classes and template functions that provide a generic way of programming. It is a library of container classes, algorithms, and iterators.

It is commonly used for efficiently programming data structures, algorithms, and functions. Some built-in data structures include arrays, vectors, queues, etc.

By – Mohammad Imran

Standard Template Library

Components

There are three major components of STL in C++:

- ✓ Containers
- ✓ Iterators
- ✓ Algorithms

By – Mohammad Imran

STL Components

Containers

Containers Library in STL gives us the Containers, which in simplest words, can be described as the objects used to contain data or rather collection of object. Containers help us to implement and replicate simple and complex data structures very easily like arrays, list, trees, associative arrays and many more.

The containers are implemented as generic class templates, means that a container can be used to hold different kind of objects and they are dynamic in nature!

By – Mohammad Imran

STL Components

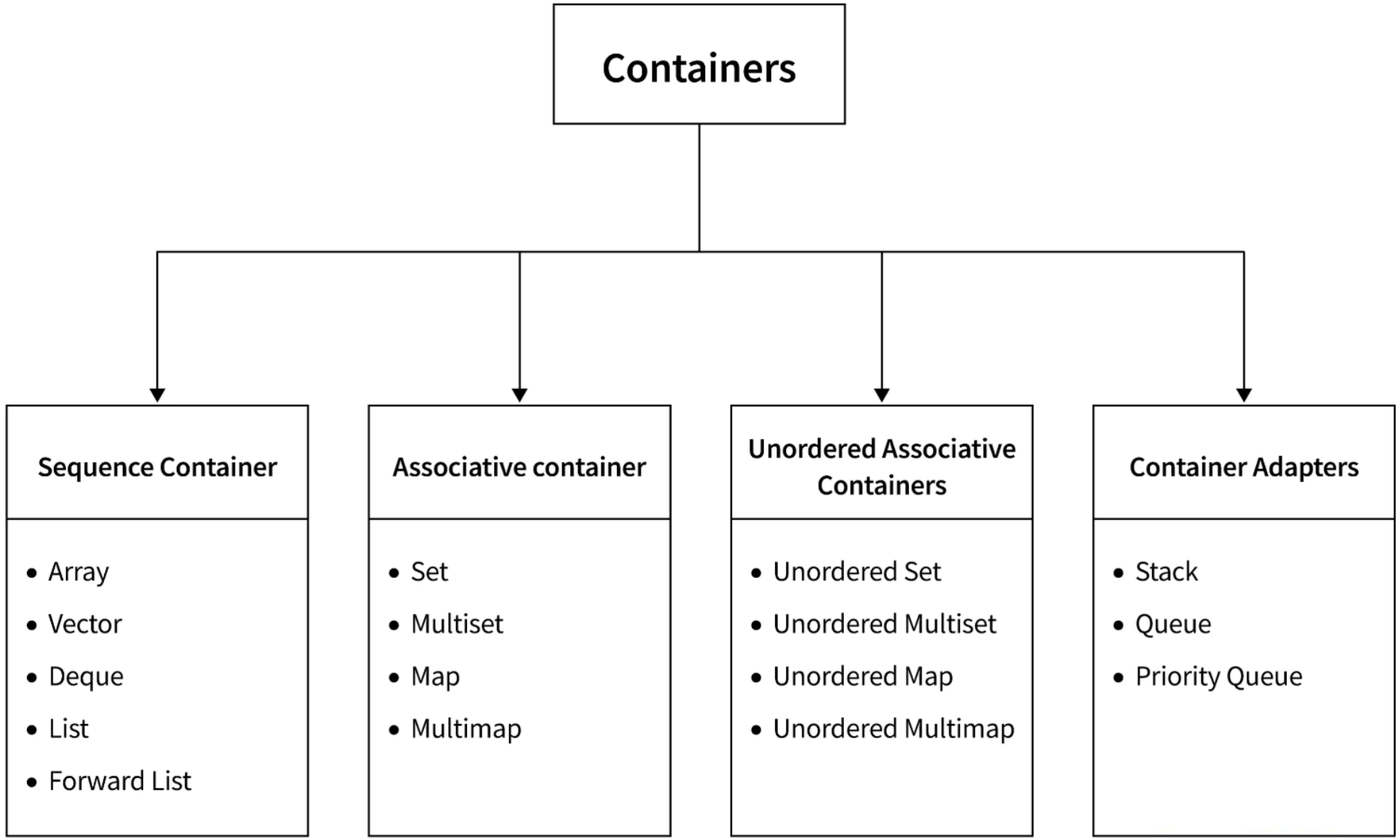
Containers

- ✓ A container is a holder object that stores other objects as elements. It is used to implement different data structures.
- ✓ They are implemented as class templates (templates use data types as parameters), allowing greater flexibility in the types supported as elements.
- ✓ The containers manage storage space for its elements and provide member functions to access them directly or through iterators.
- ✓ They replicate the most commonly used data structures like queues (queue), dynamic arrays (vector), stacks (stack), linked lists (list), queues (queue), heaps (priority_queue), trees (set), and many others.

By – Mohammad Imran

STL Components

Containers



By – Mohammad Imran

STL Components

Containers

There are 4 types of containers of STL in C++:

- ✓ **Sequence Containers** - Array, Vector, Deque, List, Forward List
- ✓ **Associative Containers** - Set, Multi-Set, Map, Multimap
- ✓ **Unordered Associative Containers** - Unordered Set, Unordered Multiset, Unordered Map, Unordered Multimap
- ✓ **Container Adapters** - Stack, Queue, Priority Queue

By – Mohammad Imran

Sequence Container

Array

Arrays, as we all know, are collection of homogenous objects. array container in STL provides us the implementation of static array, though it is rarely used in competitive programming as its static in nature but we'll still discuss array container cause it provides some member functions and non-member functions which gives it an edge over the array defined classically like

```
int array_name[array_size]
```

By – Mohammad Imran

Sequence Container

Array

Syntax:

```
array <object_type, array_size> array_name;
```

Example:

```
array <int, 5> numbers;
```

The above code creates an empty array of object_type with maximum size of array_size. However, if you want to create an array with elements in it, you can do so by simply using the = operator

```
#include <iostream>
#include <array>
using namespace std;

int main()

{

   array <int, 4> numbers = { 2, 4, 6, 8 };

   for(int i = 0; i < numbers.size(); i++)

      cout << numbers[i] << " ";

   return 0;

}
```

OUTPUT

```
2 4 6 8
```

By – Mohammad Imran

at Function

This method returns value in the array at the given range. If the given range is greater than the array size, `out_of_range` exception is thrown. Here is a code snippet explaining the use of this operator :

Array Template at Function Program - 2

```
#include <iostream>
#include <array>
using namespace std;

int main()

{

   array <int, 4> numbers = { 2, 4, 6, 8 };

   cout << numbers.at(0) << endl;

   cout << numbers.at(2) << endl;

   return 0;

}
```

OUTPUT

2
6

By – Mohammad Imran

[] Operator

The use of operator `[ ]` is same as it was for normal arrays. It returns the value at the given position in the array. Example : In the above code, statement **`cout << array1[5];`** would print 6 on console as 6 has index 5 in array1.

Array Template

[] Operator

Program - 3

```
#include <iostream>
#include <array>
using namespace std;
int main()
{
   array <int, 4> numbers = { 2, 4, 6, 8 };
   cout << numbers.at(0) << endl;
   cout << numbers[3] << endl;
   return 0;
}
```

OUTPUT

```
2
8
```

By – Mohammad Imran

front() Function

This method returns the first element in the array. As in simple array we need to write the code to find the first element of an array.

Array Template | front() Function | Program - 4

```
#include <iostream>
#include <array>
using namespace std;
int main()
{
   array <int, 4> numbers = { 2, 4, 6, 8 };
   cout << numbers.at(0) << endl;
   cout << numbers.front();
   return 0;
}
```

OUTPUT

```
2
2
```

By – Mohammad Imran

back() Function

This method returns the first element in the array. As in simple array we need to write the code to find the first element of an array.

Array Template | back() Function | Program - 5

```
#include <iostream>
#include <array>
using namespace std;

int main()

{
   array <int, 4> numbers = { 2, 4, 6, 8 };
   cout << numbers.at(0) << endl;
   cout << numbers.front() << endl;
   cout << numbers.back();
   return 0;

}
```

OUTPUT

```
2
2
8
```

By – Mohammad Imran

fill() Function

This method assigns the given value to every element of the array,

```
Example: myarray.fill(3);
```

This will fill the array myarray with value as 3, at all position.

Array Template

fill() Function

Program - 6

```
#include <iostream>
#include <array>
using namespace std;

int main()

{
   array <int, 4> arr;
   arr.fill(2);
   cout << arr.at(0) << endl;
   arr.fill(3);
   cout << arr.at(1) << endl;
   cout << arr.back() << endl;
   return 0;

}
```

OUTPUT

```
2
3
3
```

By – Mohammad Imran

swap() Function

This method swaps the content of two arrays of same type and same size. It swaps index wise, thus element of index **i** of first array will be swapped with the element of index **i** of the second array, and if swapping any of the two elements thows an execption, swap() throws exception.

Array Template

swap() Function

Program - 7

```
#include <iostream>
#include <array>
using namespace std;
int main()
{
   array <int, 8> a = {1, 2, 3, 4, 5, 6, 7, 8 };
   array <int, 8> b = {8, 7, 6, 5, 4, 3, 2, 1 };
   a.swap(b);
   cout << "a is : ";
   for(int i=0; i < 8; i++)
   {
      cout << a[i] <<" ";
   }
   return 0;
}
```

OUTPUT

```
8 7 6 5 4 3 2 1
```

By – Mohammad Imran

Operators

All operators like

==, <, >, !=, >=, <=

can be used to lexicographically compare values of two arrays.

Array Template | == Operator | Program - 8

```
#include <iostream>
#include <array>
using namespace std;
int main()
{
   array<int, 4> array1 = {2, 4, 6, 8};
   array<int, 4> array2 = {2, 4, 6, 8};
   array<int, 4> array3 = {1, 3, 5, 7};
   if (array1 == array2)
   {
      cout << "array1 and array2 are equal." << endl;
   }
   else
   {
      cout << "array1 and array2 are not equal." << endl;
   }
```

OUTPUT

```
8 7 6 5 4 3 2 1
```

By – Mohammad Imran

Array Template | == Operator | Program - 8

```
    if (array1 == array3)
    {
       cout << "array1 and array3 are equal." << endl;
    }
    else
    {
       cout << "array1 and array3 are not equal.";
    }
    return 0;
}
```

OUTPUT

```
array1 and array2 are equal.
array1 and array3 are not equal.
```

By – Mohammad Imran

Coding Questions

By – Mohammad Imran

Question - 1

Write a C++ operational program for file handling according to the given menu.

```
---------MENU----------
1. Create File
2. Write Data
3. Read Data
4. Append Data
5. Delete Data
6. Search Data
7. Update Data
0. Exit
-----------------------
Enter Your Choice :
```

Click here to see code

By – Mohammad Imran

QUIZ

By – Mohammad Imran

Quiz - 1

```
#include <iostream>                    //If file Data.txt not
#include <fstream>                     //found then output is
using namespace std;
int main()
{
   ofstream outfile;
   outfile.open("Data.txt",ios::app);
   if(!outfile)
   {
      cout << "File Not Found" << endl;
      return -1;
   }
   outfile << "Hello World" << endl;
   return 0;
}
```

OUTPUT

Data will inserted in the file

By – Mohammad Imran