Abstraction and Paradigms of Programming

Bachelor of Engineering, Computer Engineering

II nd Year IV th Semester

Iterators in STL

An iterator is used to point to the memory address of the STL container classes. For better understanding, you can relate them with a pointer to some extent.

They allow you to iterate over the container, access and assign the values, and run different operators over them, to get the desired result.

Use of Iterators in C++

An iterator in C++ serves the following major purposes:

- The primary objective of an iterator is to access the STL container elements and perform certain operations on them.
- The internal structure of a container does not matter, since the iterators provide common usage for all of them.
- Iterator algorithms are not dependent on the container type.
- An iterator can be used to iterate over the container elements. It can also provide access to those elements to modify their values.
- Iterators follow a generic approach for STL container classes. This way, the programmers don’t need to learn about different iterators for different containers.

Syntax of Defining Iterators

<Container_Type> :: iterator iterator_name;

Using Iterators in C++

Before Iterators

```
Int main(){
Vector<int> v1 = {2,3,5,6,7};
For(int i=0;i<v1.size();I++)
Cout<<v[i]<<endl;
}
```

With Iterators

```
Vector<int>::iterator it=v.begin();
Cout<<(*it)<<endl;
Cout<<(*(it+1))<<endl;
```

Using iterators with containers that have no indexing

```
Vector<int>::iterator it = v.begin();
For(it=v.begin();it!=v.end();it++){
Cout<<(*it)<<endl;
}
```

Continuity and Discontinuity in Iterators

It++ -- Next element in the container.

It+1 – Next location in the container.

Accessing a vector of pairs using iterators

```
Vector<pair<int,int>>v_p={{1,2},{2,3},{3,4}}
Vector<pair<int,int>>::iterator it;
For(it=v_p.begin();it!=v_p.end();it++){
Cout<<(*it).first<<endl;
Cout<<(*it).second<<endl;
}
```

Maps in C++ STL