Summary

Sessions (17-01-2023)

- A **container** is an object that stores a collection of objects of a specific type. For example, if we need to store a list of names, we can use a vector.

- In Data Structure if we use an array then we can access the array elements in constant time. But if we want to insert an element or delete an element then this type of operation needs lots of time in an array. Therefore they need n cpu time.

- If we talk about RAM then reading data from any memory address is the same/constant for all. But if we talk about Storage then reading data from different-different addresses is vary.

- For traversing into any data structures we have to use different-different code. But STL created some standards so that we use the same for all data structures and that is known as iteration. For this, an iterator will be used.

- Iterators are used to point at the memory addresses of STL containers. An iterator is an object (like a pointer) that points to an element inside the container. We can use iterators to move through the contents of the container.

- An Iterator can hold an address and if we want to traverse into any data structure using an iterator then we need some kind of method which will return some beginning of the data structure and ending of the data structure.

  Operations of iterators :

  1) **begin()** : This function is used to return the beginning position of the container.

2) **end()** : This function is used to return the after end position of the container.

- Example :

```
// Declaring iterator to a vector
vector<int>::iterator itr;

for (itr = v1.begin(); itr != v1.end(); itr++) {
    cout << *itr << endl;
}
```

- In new version of C++ we can use some other method for traversing ie. foreach loop.

```
//foreach loop
for ( int x:v1) {
    cout << x << endl;
}
```