

vector(STL) vs Vector(SPL)

There isn't much significant difference between the two except the way they insert and delete elements. We would be first trying to talk about vector(STL) methods then you need to correlate it to Vector(SPL) that you have studied in class.

vector(STL) methods

1. **push_back(item)** // similar to **add(value)** of Vector(SPL)

This method is used to add an element to the end of the list.

Example:-

```
vector<int> num; // definition of the vector
num.push_back(5); // now num = [5]
num.push_back(9); // now num = [5,9]
```

You can clearly see that 9 got added just after the element 5.

2. **erase(iterator)** // similar to **remove(i)** of Vector(SPL)

This method is used to delete an element from a specified location. But note that this method takes iterator as argument instead of index value. For understanding purpose, think of iterator as a variable that store the address of a location instead of its index value.

Example:-

```
vector<int> num; // definition of the vector
num.push_back(5); // now num = [5]
num.push_back(9); // now num = [5,9]
num.push_back(2); // now num = [5,9,2]
num.erase(num.begin() + 1) // now num = [5,2], you can see the
element 9 which was at index 1 got deleted.
num.push_back(4); // now num = [5,9,4]
num.push_back(1); // now num = [5,9,4,1]
num.erase(num.begin() + 3) // now num = [5,9,4], you can see the
element 1 which was at index 3 got deleted.
```

3. insert(iterator, value) //similar to insert(i, value) of Vector(SPL)

This method inserts an element at the specified index. But again it takes iterator instead of index value.

Example:-

```
vector<int> num; // definition of the vector
num.push_back(5); // now num = [5]
num.push_back(9); // now num = [5,9]
num.push_back(2); // now num = [5,9,2]
num.insert(num.begin() + 1, 7); // now num = [5,7,9,2], you can
clearly see that the element 7 got added at location 1 and all the
elements got shifted to the right
num.insert(num.begin() + 3, 4); // now num = [5,7,9,4,2], you can
clearly see that the element 4 got added at location 3 and the
element 2 got shifted to the right.
```

4. at(index) //similar to get(i) of Vector(SPL)

This method simply returns the element at the specified index.

Example:-

```
vector<int> num; // definition of the vector
num.push_back(5); // now num = [5]
num.push_back(9); // now num = [5,9]
num.push_back(2); // now num = [5,9,2]
cout<<num.at[0]; // prints 5, as 5 is at location 0.
cout<<num.at[1]; // prints 9, as 9 is at location 1.
cout<<num.at[2]; //prints 2, as 2 is at location 2.
```

Looping Constucts:-

There are 2 ways you can loop through all the elements of the vector.

1. using index value

Example:-

```
for(int i=0; i< num.size(); i++){
cout<<num.at(i)<<endl;
}
```

or

```
for(int i=0; i< num.size(); i++){
cout<<num[i]<<endl;
}
```

2. using iterator

```
for(vector<int>::iterator it = num.begin(); it != num.end(); it++){  
    cout<<*it<<endl;  
}
```