**LIST :**

They are same type of containers like the the vectors and perform the same functionality as the vectors like:

1. **CREATE LIST:**

**Syntax: list<data\_type> variabel\_name;**

1. **PUT VALUES IN AN EMPTY LIST:**

**Syntax: list<int> a;**

**a.push\_back();**

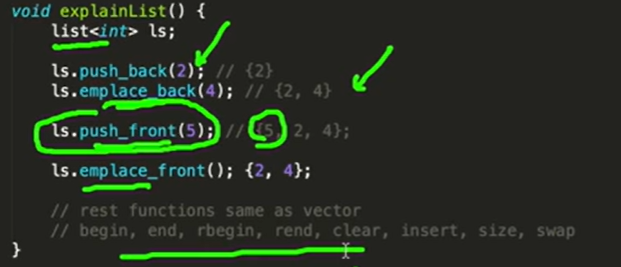
**a.emplace\_back();**

**Now unlike the vector we can only put values from the back in list we can do it from the front**

**a.push\_front();**

**a.emplace\_front();**

**Rest functions like insert(),erase() and etc are same for list and vector.**

****

**Note**: when we try to print elements in list:

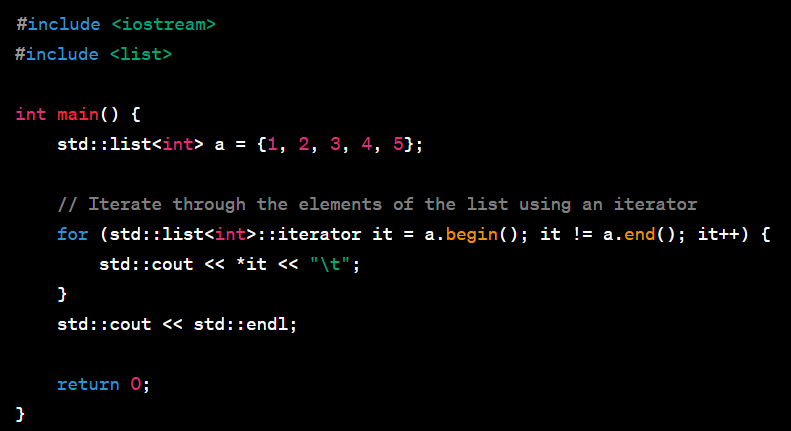
for(int i=0;i<a.size();i++){

cout<<a[i]<<"\t";

}

cout<<endl;

This is not allowed as Unlike arrays and vectors, lists do not provide random access to elements using the [] operator. Instead, you can use an iterator to traverse the elements of the list. Here's an example:



We have to use an iterator for the list .

**Note: insert() / erase() in vector we could give “v.begin()/end()/etc” as iterator instead of it but in list we have to use iterator:**

**Example:**

**int main(){**

**list <int> a;**

**a.push\_front(2);**

**a.push\_back(3);**

**a.push\_back(4);**

**a.push\_back(5);**

**auto it=a.begin();**

**advance(it,2);**

**a.insert(it,88);**

**for(auto i:a){**

**cout<<i<<"\n";**

**}**

**}**

**LETS SEE FOR SWAP OPERATION:**

**Syntax: l1.swap(l2);**

Explanation: if **list<int> l1={2,3,4} list<int> l1={7,8,9}**

**l1.swap(l2);----------->> l1{7,8,9} l2{2,3,4}**

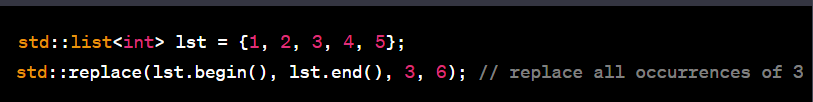
**List acts like a doubly linked list hence we can use the push\_front() unlike the vector if we have to insert the element from the front we have to use insert() which has higher time complexity.**

**Vector are singly linked list.**

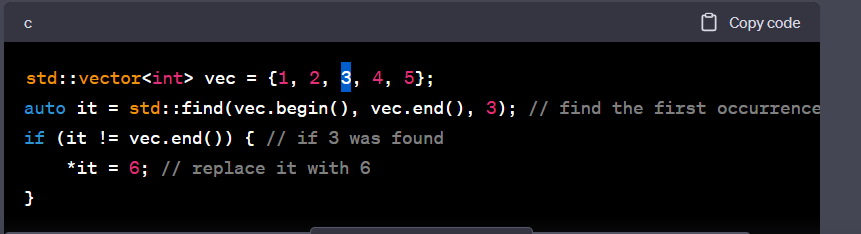
**Use containers accordingly.**

**To replace an element in a list/vector is done:**

1. **To replace all occurrence of the element:**

****

**2)to replace the specific element with other:**

****

List does’t store value in a continues location hence we need iterators in vectors values are stored in a continues location so we can directly use funtaniolity like being()+1 in erase/insert.