```cpp
#include<bits/stdc++.h> //Header file for STL
using namespace std;
int main()
{
    cout<<"Hello STL VECTOR!"<<endl<<endl;
    vector<int>v; //Vector declaration(Vector is a dynamic array).
    //Store data to vector using push_back() that always store data at the end.
    v.push_back(1); //Store 1 at index 0 / v[0]
    v.push_back(2); //Store 2 at index 1 / v[1]
    v.push_back(3); //Store 3 at index 2 / v[2]
    //Print data of vector
    cout<<v[0]<<" "<<v[1]<<" "<<v[2]<<endl; //General method - print garbage value if the index does not exist.
    cout<<v.at(0)<<" "<<v.at(1)<<" "<<v.at(2)<<endl; //Using at() function - print error message if the index does not exist.
    cout<<"Vector Size: "<<v.size()<<endl; //Get the size of the vector.
    v.front()=5; //Replace/Assign the first value of vector using front() function.
    v.back()=10; //Replace/Assign the last value of vector using back() function.
    for(int i=0;i<v.size();i++){
        cout<<v.at(i)<<" "; //Print vector using v.size()
    }
    cout<<endl;
    cout<<v.front()<<endl; //Print the first value of vector using front() function.
    cout<<v.back()<<endl; //Print the last value of the vector using back() function.
    v.clear(); //Clear the vector or delete all data of vector using clear() function.
    cout<<v.size()<<endl; //Size is 0 after clearing the vector means there is no data in vector.

    //Check weather the vector is empty or not using empty() function.
    if(v.empty()) cout<<"Empty"<<endl;
    else cout<<"Not Empty"<<endl<<endl;
    vector<int>vv = {1,2,3,4,5}; // Another process to store data to vector.
    for(int i=0;i<vv.size();i++){
        cout<<vv.at(i)<<" ";
    }
    cout<<endl;
    vv.pop_back(); //Delete the last data of vector using pop_back() function.
    for(int i=0;i<vv.size();i++){
        cout<<vv.at(i)<<" ";
    }
    cout<<endl;
    vv.erase(vv.begin()+2); //Delete the data of index 2 using erase() function.
    cout<<"vv before swap : ";
    for(int i=0;i<vv.size();i++){
        cout<<vv.at(i)<<" ";
    }
    cout<<endl;
    vector<int>vvv = {1,2,3,4,5};
    vvv.erase(vvv.begin()+2,vvv.end()); //Delete data from index 2 to last index.
    for(int i=0;i<vvv.size();i++){
        cout<<vvv.at(i)<<" ";
    }
    cout<<endl;
    vvv.insert(vvv.begin()+1,5); //Insert a data in a certain index(1) using insert() function.

    for(int i=0;i<vvv.size();i++){
        cout<<vvv.at(i)<<" ";
    }
    cout<<endl;
    vvv.insert(vvv.begin()+1,3,5); //Insert data multiple times.
    cout<<"vvv before swap : ";
    for(int i=0;i<vvv.size();i++){
        cout<<vvv.at(i)<<" ";
    }
    cout<<endl;
    swap(vv,vvv); // Swap two vector using swap() function.
    cout<<"vv after swap and before sorting: ";
    for(int i=0;i<vv.size();i++){
        cout<<vv.at(i)<<" ";
    }
    cout<<endl;
    cout<<"vvv after swap : ";
    for(int i=0;i<vvv.size();i++){
        cout<<vvv.at(i)<<" ";
    }
    cout<<endl;
    sort(vv.begin(),vv.end()); //Sort the elements of a vector increasingly.
    cout<<"vvv after sorting : ";
    for(int i=0;i<vv.size();i++){
        cout<<vv.at(i)<<" ";
    }
    cout<<endl;
    reverse(vv.begin(),vv.end()); //Reverse a vector.
    cout<<"vvv after reverse : ";
    for(int i=0;i<vv.size();i++){
        cout<<vv.at(i)<<" ";
    }
    cout<<endl;
    vector<int>::iterator it; //Declaring iterator
    for(it=vv.begin();it!=vv.end();it++){
        cout<<*it<<" "; //Print a vector using iterator
    }
    cout<<endl;
    return 0;
}
```