**Gr no 21810819**

**Assignment No:** 1

**Problem Statement:**

Create a class template to represent a generic vector. Include the following member functions:

1. To create the vector.
2. To modify the value of a given element.
3. To multiply by a scalar value.
4. To display the vector in the form(10,20,30,….).

**Aim of Assignment:**

To implement the vector concept in C++ and perform various operations on vector.

**Description:**

A generic integer vector is defined. A iterator object is used to iterate through the vector. Five functions are defined to perform various operations on vector such as accept, display, insert at given position, modify and multiply by given quantity. Each function is called in switch. Program is made menu driven to perform the task as per user’s choice.

**OOP Concept used:**

1. Vector: - Vectors are same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container. Vector elements are placed in contiguous storage so that they can be accessed and traversed using iterators. In vectors, data is inserted at the end. Inserting at the end takes differential time, as sometimes there may be a need of extending the array. Removing the last element takes only constant time because no resizing happens. Inserting and erasing at the beginning or in the middle is linear in time

Functions in vector-

1. [begin()](https://www.geeksforgeeks.org/vectorbegin-vectorend-c-stl/) – Returns an iterator pointing to the first element in the vector
2. [size()](https://www.geeksforgeeks.org/vectorempty-vectorsize-c-stl/) – Returns the number of elements in the vector.
3. [at(g)](https://www.geeksforgeeks.org/vectorat-vectorswap-c-stl/) – Returns a reference to the element at position ‘g’ in the vector
4. push\_back()- function is used to push elements into a vector from the back. The new value is inserted into the vector at the end, after the current last element and the container size is increased by 1.
5. Iterator:- Iterators are used to point at the memory addresses of [STL](http://quiz.geeksforgeeks.org/the-c-standard-template-library-stl/) containers. They are primarily used in sequence of numbers, characters etc. They reduce the complexity and execution time of program

**Code :**

#include<iostream>

#include<vector>

using namespace std;

void display(vector<int> &v)

{

cout<<"(";

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

{

cout<<v[i];

if(i!=v.size()-1)

cout<<",";

}

cout<<")";

cout<<"\n";

}

int main()

{

vector<int> v; // creating vector object

cout<<"Initial size "<<v.size()<<endl; // initial size of vector using size()

int x,no,ino,dno,pi,ie,de,scr,size=0;

cout<<"Enter how many elements you want? ";

cin>>no;

cout<<"Enter values "<<endl;

for(int i=0;i<no;i++) // Accepting vector using push\_back()

{

cin>>x;

v.push\_back(x);

}

cout<<"\nSize after inserting "<<no<<" values= "<<v.size();

cout<<"\nCurrent contents= "<<endl;

display(v);

cout<<"Scalar quantity"; // Scalar quantity

cout<<"\nEnter Scalar quantity= ";

cin>>scr;

vector<int>::iterator itr4=v.begin(); // iterator object

for(int j=0;j<no;j++)

v[j]=v[j]\*scr;

display(v);

cout<<"Enter element to push in vector= ";

cin>>pi; // Displaying vector

v.push\_back(pi);

cout<<"\nSize after push\_back()= "<<v.size()<<endl;

cout<<"\nNow the contents of vector after push\_back()"<<endl;

display(v);

cout<<"Insertion"<<endl;

cout<<"\nEnter element position= ";

cin>>ino;

vector<int>::iterator itr1=v.begin(); // iterator object

itr1=itr1+ino;

cout<<"Enter Number to insert= "<<endl;

cin>>ie;

v.insert(itr1,1,ie); // Iterator created and inserted

cout<<"\nContents of vector after insrting= ";

display(v);

cout<<"\nRemoving elements position= "<<endl;

cout<<"Enter position of removing element= ";

cin>>de;

v.erase(v.begin()+de); // Removing elements

cout<<"\nContents after deletion= ";

display(v);

cout<<endl;

return 0;

}

**Conclusion:** The various operations on vector are implemented successfully.

**Screenshots** :

