**//vectorDemo.cpp**

**//A program to demonstrate the Vector class**

**//Author: nmessa**

#include <iostream>

#include <cstdlib> //for random number generation based on

#include <ctime> //system clock seed value

#include <vector>

using namespace std;

void printVector(vector<int>);

int main()

{

vector<int> vect(10);

int number, i;

int temp;

srand(time(0)); //initialize pseudorandom number generator

//Fill vector with random numbers

for (i = 0;i < vect.size(); i++)

vect[i] = rand();

printVector(vect);

//Add a number to the vector

cout << "\nI just added this number (12345): ";

vect.push\_back(12345);

printVector(vect);

//Pop an element from the vector

temp = vect[vect.size()-1];

vect.pop\_back();

cout << "\nPopped value = " << temp << endl;

printVector(vect);

cout << "\nI am re-sizing the array" << endl;

vect.resize(20);

cout << "The vector now has " << vect.size() << " elements" << endl;

printVector(vect);

return 0;

}

//This function takes a Vector class object and displays it in a visual format

void printVector(vector<int> v)

{

int i;

cout << "The vector contains the following data: " << endl;

for (i = 0;i < v.size()- 1; i++)

{

if (i % 5 == 0)

cout << endl;

cout << v[i] << " ---> ";

}

cout << v[i] << endl;

}

**//Vector class demo showing the use of an iterator**

**//also demonstrates the passing of a vector to a function**

**//by value and by reference**

#include <iostream>

#include <vector>

using namespace std;

#define SIZE 10

void displayVector(vector<int>);

void reverseVector(vector<int> &);

int main ()

{

vector<int> myVector;

for (int i = 1; i <= SIZE; i++) //Fill the vector

myVector.push\_back(i);

cout << "myVector contains:" << endl; //Display vector

displayVector(myVector);

//Display the vector forward using an iterator

cout << "Displaying the vector forwards using an iterator ....." << endl;

vector<int>::iterator it;

for (it = myVector.begin(); it < myVector.end(); it++)

cout << \*it << " ";

cout << endl << endl;

//Display the vector backwards using an iterator

cout << "Displaying the vector backwards ....." << endl;

vector<int>::reverse\_iterator rit;

for ( rit=myVector.rbegin() ; rit < myVector.rend(); ++rit )

cout << " " << \*rit;

cout << endl;

//Pass the vector to a function to reverse it

cout << "Reversing the vector ....." << endl;

reverseVector(myVector);

cout << "The vector now contains: " << endl;

displayVector(myVector);

return 0;

}

void displayVector(vector<int> nums)

{

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

cout << nums[i] << " ";

cout << endl;

}

void reverseVector(vector<int> &nums)

{

vector<int> bNums; //Create temporary vector

bNums.resize(nums.size()); //Make the same size

for (int i = 0; i < nums.size(); i++) //copy vector backwards

bNums[bNums.size()-i-1] = nums[i];

for (int i = 0; i < nums.size(); i++) //copy backwords vector

nums[i] = bNums[i]; //into original vector

}