**Data Structures and Algorithms**

**Lab-12**

**Name:** Ahmad Amjad Mughal

**Reg No:** 121672

**Class:** BSCS-6C

**Task:**

//bubble sort implementation in C++

//Ahmad Amjad Mughal

//121672

//BSCS-6C

#include <iostream>

#include <vector> //STL data structure that have built in functions

namespace{

typedef unsigned int uint;

}

std::vector<int> sort(std::vector<int>a) //user defined function of vector type that returns int

{

for (int i = 1; i < a.size(); ++i){ //outer loop

for (int j = 0; j < a.size() - 1; ++j){ //inner loop

if (a[j] > a[i]) std::swap(a[j], a[i]); //comparison of elements of consective locations we use built in swap function

}

}

return a; //returns sorted array

}

int main()

{

std::vector<int> a = { 17, 12, 1, 20, 4, 6, 94 }; //array defined of vector type

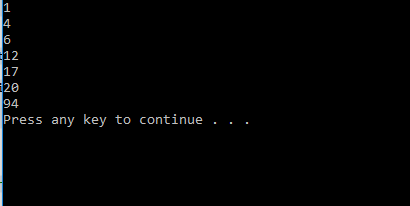
for (uint i = 0; i < a.size(); ++i){

std::cout << sort(a)[i] << std::endl; //function call

}

}

**Screenshot:**

****

**Explanation:**

I use vector built-in functions to sort a random array. Bubble sort is the simplest one to sort the array. We declare array of vector type and pass it to another user defined function of vector class that performs sorting. It calls the swap function that is built-In and performs swapping when certain condition meet.