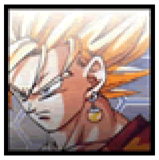
Vectors in C++



Contributed by blitzcoder On 2nd September, 2010
This is an article on Vectors in C++ **in** C++.
We have all read about C++ Standard Template Library. It provides generic code for data types. Like vectors, pairs, list, sets, maps, etc. Today we'll take a good look into Vectors.

VECTOR

We have all been using arrays to store elements. But you must have thought at one point or another, "Man! I am wasting a lot of space". This is where C++ STL's vector comes into use. If you are not sure about the size of the array you want to use, you should use a vector.

The first thing you need to do include "vector"

Code:

```
#include<vector>
using namespace std;
```

Now let us declare a 1-D integer vector,

Code:

```
vector<int> s;  //vector with no elements
vector<int> s(10); //vector with 10 elements all set to zero
vector<int> s[10]; //an array of 10 vectors each with no elements.
```

(note the usage of square brackets and parentheses)

So now that we know how to declare a 1-D vector, we should realize the fact that "int" can be replaced by any data type since vector is a template. So we can make char, string, float, double, etc. types of vectors.

Now we will see how to work on vectors. Here is a small snippet of code --

Code:

```
vector<int> v(10);
for(int i=0;i<10;i++)
{
 v[i]=i;
}
v.push_back(2);
```

What we have done in the code above is that we have made a vector of 10 elements, filled it up and then used the inbuilt function "push_back" to add 2 to the end of the vector. Pretty cool, eh? We have thus changed the size of the vector as well, from 10 to 11.

The most frequently used feature of vector is that it can report its size.

Code:

```
 int elements_count = v.size();
```

Two things you should remember: first, size() is unsigned, which may sometimes cause problems. Accordingly, I usually define macros, something like sz(C) that returns size of C as ordinal signed int. Second, it's not a good practice to compare v.size() to zero if you want to know whether the container is empty. You're better off using empty() function:

Code:

```
bool is_nonempty_notgood = (v.size() >= 0); // Try to avoid this
 bool is_nonempty_ok = !v.empty();
```

This is because not all the containers can report their size in O(1), and you definitely should not require counting all elements in a double-linked list just to ensure that it contains at least one.

Now we will see how to use the resize function (again an inbuilt one) to change the size of a vector.

Code:

```
//assuming that we have a predeclared vector of 10 elements
v.resize(15); //adds 5 element space to the end of the vector
v.resize(6);   //deletes the last 4 elements
```

A very useful function to use is clear(). This function removes all the elements from a vector, thus, effectively setting its size to 0. You can use erase() to erase elements from the end one by one.

You can also assign values to vectors while declaring them,

vector<string> s(30,"Hello World"); //makes a vector of 30 elements and fills in the first 11 elements

The next and last most important thing to remember is that when you pass a vector to a function, it creates a new copy of the vector which is not preferred. Thus when receiving the vector the format should be as follows,

Code:

```
void fn(vector<int> &v)
```

Thus we are getting the reference of the original vector and work on it only.

All the above functionalities can be applied to a multi-dimensional vector as well. (Yes, you can make multi-dimensional vectors too! 🙂)

Code:

```
 vector< vector<int> > Matrix;                                     //no elements added
 vector< vector<int> > Matrix(N, vector<int>(M, -1));   //NxM array with all elements assigr
```

Now equipped with the power of STL (vector, for now) you can make your program much more optimized!

Related Articles

- 100 Multiple choice questions in C, by **coderzone** in C
- Overview of C#, by **Sanskruti** in C#
- 40 SEO Interview Questions, by **coderzone** in Internet Marketing