Table of Contents

C++ Notes

Const and Pointers

```
char* const cp;   // const pointer to char
char const* pc;   // pointer to const char
const char* pc2;  // pointer to const char
```

Deleting elements from std::list

Typically, the default STL container one should use is `vector` because it allows one to have random access to its elements (*e.g.* `vec[5]`), but it is slow if you need to remove or insert entries in the middle of the `vector`. In this case, one should use a `list`. If you loop over a `list` to remove elements satisfying some criteria, you have to be careful with the state of iterators after an element has been erased, because the erased iterator become invalid. To get around this, you can note the difference between the pre increment (++it) and the post increment (it++). The post increment actually returns a copy of the iterator before it was incremented. This way you can safely step the iterator along to the next element before calling `erase`.

```
std::list<Muon>::iterator muon_it;
for(muon_it=muon_list.begin(); muon_it!=muon_list.end();)
{
    if(!( muon_it->get_matchChi2OverDoF() < 8.0 ))
    {
        muon_list.erase(muon_it++); // note that post increment (it++) increments the iterator be
    }
    else
    {
        ++muon_it;
    }
}
```

Descending Sort

```
#include <algorithm> // for sort
#include <functional> // for greater

std::vector<std::pair<float, int> > et_sorted_indices;
for(unsigned int i_cluster=0; i_cluster < (*m_tau_cluster_n)[i_tau]; i_cluster++)
{
    et_sorted_indices.push_back(std::pair<float, int>((*m_tau_cluster_E)[i_tau][i_cluster], i_clu
}
std::sort( et_sorted_indices.begin(), et_sorted_indices.end(), std::greater<std::pair<float, int>
```

Note that pairs are sorted by the first object in the pair. See greater here.

Calculating Delta Phi

```
#include <cmath>

float delta_phi(float phi1, float phi2)
{
    return fmod(phi1 - phi2 + 3*M_PI ,2*M_PI) - M_PI;
}
```

-- RyanReece - 24 Sep 2008 -- RyanReece - 24 Apr 2009

This topic: Sandbox > CppNotes
Topic revision: r6 - 2010-03-16 - RyanReece

  Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback