```
// File and Version information:
// $Header: /nfs/slac/g/glast/ground/cvs/Event/src/Recon/CalRecon/CalCluster.cxx,v 1.5
2002/06/12 00:16:44 chehtman Exp $
//
//  Implementation file of CalCluster and CalClusterCol classes
//
// Authors:
//
//    Alexandre Chekhtman
//    Regis Terrier
//    Jose Angel Hernando
//
//

#include "Event/Recon/CalRecon/CalCluster.h"

using namespace Event;

CalCluster::CalCluster(double e,Point p)

// Purpose: constructor with parameters
//
// Inputs:
//   e - energy sum
//   p - cluster position
//
{
   int nl = 8;
```

Can we avoid this hard coded value for the number of layers?

```
   // set vectors length to the number of layers
   m_eneLayer.resize(nl);
   m_pLayer.resize(nl);

   // reset all data members to 0
   ini();

   //set energy sum
   m_energySum = e;

   // temporary this data member is set to raw energy sum
   m_energyCorrected = m_energySum;

   //set position
   m_position = p;
}
```

```
void CalCluster::writeOut(MsgStream& stream) const

// Purpose: provide ascii output of some data members for
//          debugging purposes
// Input:
//        stream - Gaudi message stream
{
    stream << "Energy " << m_energySum
        << " Corrected " << m_energyCorrected;
    stream << " " << getPosition().x()
        << " " << getPosition().y()
        << " " << getPosition().z();
    stream << " " << getDirection().x()
        << " " << getDirection().y()
        << " " << getDirection().z();
    stream << endreq;
}



void CalCluster::ini()
// Purpose: reset all data members to 0
//

{
    m_energySum       = 0.;
    m_energyCorrected = 0.;

    m_position = Point(0.,0.,0.);
    m_direction = Vector(0.,0.,0);
    int nLayers = m_eneLayer.size();
    for(int i = 0; i<nLayers; i++){
        m_eneLayer[i]=0.;
        m_pLayer[i]=Vector(0.,0.,0.);
    }
}

void CalClusterCol::delClusters()

//Purpose: delete all CalCluster object from memory

{
    int nClusters = num();
```

```
   for (int icl = 0; icl < nClusters; icl++) {
      delete operator[](icl);
   }
   clear();
}

void CalClusterCol::ini()

//Purpose:  delete all pointers to clusters
// from collection

{
   clear();
}


void CalClusterCol::writeOut(MsgStream& stream) const

// Purpose: provide symbolic output of some data members
//         of all clusters in collection for debugging purposes
//
// Input:
//       stream - Gaudi message stream
{

   // if there is no clusters - return
   if (size()<=0) return;

   stream << " --- CalClusterCol  --- " << size() << endreq;

   // loop over all clusters
   for (int i = 0; i < size();i++) {

      // call the writeOut() method for each cluster
      (operator[](i))->writeOut(stream);
   }

}
```