```
//
// ********************************************************************
// * License and Disclaimer                                           *
// *                                                                  *
// * The  Geant4 software  is  copyright of the Copyright Holders  of *
// * the Geant4 Collaboration.  It is provided  under  the terms  and *
// * conditions of the Geant4 Software License,  included in the file *
// * LICENSE and available at  http://cern.ch/geant4/license .  These *
// * include a list of copyright holders.                             *
// *                                                                  *
// * Neither the authors of this software system, nor their employing *
// * institutes,nor the agencies providing financial support for this *
// * work  make  any representation or  warranty, express or implied, *
// * regarding  this  software system or assume any liability for its *
// * use.  Please see the license in the file  LICENSE  and URL above *
// * for the full disclaimer and the limitation of liability.         *
// *                                                                  *
// * This  code  implementation is the result of  the  scientific and *
// * technical work of the GEANT4 collaboration.                      *
// * By using,  copying,  modifying or  distributing the software (or *
// * any work based  on the software)  you  agree  to acknowledge its *
// * use  in  resulting  scientific  publications,  and indicate your *
// * acceptance of all terms of the Geant4 Software license.          *
// ********************************************************************
//

#include "TrackerHit.hh"
#include "G4ios.hh"
#include "G4UnitsTable.hh"
#include <iostream>

G4Allocator<TrackerHit> TrackerHitAllocator;

TrackerHit::TrackerHit():theEntryPoint(0),theExitPoint(0){
  theMomentum = 0;
  theTimeOfFlight = 0;
  theEnergyLoss = 0;
  theParticleType = 0;
  theDetectorId= 0;
  theTrackIdInHit= 0;
  theThetaAtEntry = 0;
  thePhiAtEntry = 0;
}

TrackerHit::TrackerHit(G4ThreeVector aEntry,G4ThreeVector aExit,G4double aMom,G4
double aToF,G4double aEloss,
                       G4int aPDG ,unsigned int aID,unsigned int aTkID,G4double
aTheta,G4double aPhi)
{
  theEntryPoint = aEntry;
  theExitPoint = aExit;
  theMomentum = aMom;
  theTimeOfFlight = aToF;
  theEnergyLoss = aEloss;
  theParticleType = aPDG;
  theDetectorId = aID;
  theTrackIdInHit = aTkID;
  theThetaAtEntry = aTheta;
  thePhiAtEntry = aPhi;
}

TrackerHit::~TrackerHit()
{;}


TrackerHit::TrackerHit(const TrackerHit &right)
  : G4VHit()
{
  theEntryPoint = right.theEntryPoint;
```

```
  theExitPoint = right.theExitPoint;
  theMomentum = right.theMomentum;
  theTimeOfFlight = right.theTimeOfFlight;
  theEnergyLoss = right.theEnergyLoss;
  theParticleType = right.theParticleType;
  theDetectorId = right.theDetectorId;
  theTrackIdInHit = right.theTrackIdInHit;
  theThetaAtEntry = right.theThetaAtEntry;
  thePhiAtEntry = right.thePhiAtEntry;
}

const TrackerHit& TrackerHit::operator=(const TrackerHit &right)
{
  theEntryPoint = right.theEntryPoint;
  theExitPoint = right.theExitPoint;
  theMomentum = right.theMomentum;
  theTimeOfFlight = right.theTimeOfFlight;
  theEnergyLoss = right.theEnergyLoss;
  theParticleType = right.theParticleType;
  theDetectorId = right.theDetectorId;
  theTrackIdInHit = right.theTrackIdInHit;
  theThetaAtEntry = right.theThetaAtEntry;
  thePhiAtEntry = right.thePhiAtEntry;
  return *this;
}

G4int TrackerHit::operator==(const TrackerHit &right) const
{
  return (theEntryPoint == right.theEntryPoint &&
          theExitPoint == right.theExitPoint &&
          theMomentum == right.theMomentum &&
          theTimeOfFlight == right.theTimeOfFlight &&
          theEnergyLoss == right.theEnergyLoss &&
          theParticleType == right.theParticleType &&
          theDetectorId == right.theDetectorId &&
          theTrackIdInHit == right.theTrackIdInHit &&
          theThetaAtEntry == right.theThetaAtEntry &&
          thePhiAtEntry == right.thePhiAtEntry);
}


void TrackerHit::Draw()
{;}


void TrackerHit::Print()
{
  std::cout<<"TrackerHit = "<<std::endl;
  std::cout<<"EntryPoint = "<<theEntryPoint<<"; "<<std::endl;
  std::cout<<"ExitPoint = "<<theExitPoint<<"; "<<std::endl;
  std::cout<<"Momentum = "<<theMomentum<<"; "<<std::endl;
  std::cout<<"Time Of Flight = "<<theTimeOfFlight<<"; "<<std::endl;
  std::cout<<"Energy Loss= "<<theEnergyLoss<<"; "<<std::endl;
  std::cout<<"Particle Type = "<<theParticleType<<"; "<<std::endl;
  std::cout<<"Detector ID = "<<theDetectorId<<"; "<<std::endl;
}

void TrackerHit::SetExitPoint(G4ThreeVector aExit){
  theExitPoint = aExit;
}

void TrackerHit::AddEnergyLoss(G4double aEloss){
  theEnergyLoss += aEloss;
}
```