```cpp
//
// ********************************************************************
// * 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.          *
// ********************************************************************
//

#ifndef TrackerSD_h
#define TrackerSD_h 1

#include "G4VSensitiveDetector.hh"
#include "TrackerHit.hh"
#include <map>

class G4Step;
class G4HCofThisEvent;
class G4TouchableHistory;

class TrackerSD : public G4VSensitiveDetector
{

public:
  TrackerSD(G4String name);
  ~TrackerSD();

  void Initialize(G4HCofThisEvent* HCE);
  G4bool ProcessHits(G4Step* aStep,G4TouchableHistory* ROhist);
  G4bool NewHit(G4Step* aStep);
  int SetDetectorId(G4Step* aStep);
  void CreateHit(G4Step* aStep);
  void UpdateHit(G4Step* aStep);
  void EndOfEvent(G4HCofThisEvent* HCE);
  void clear();
  void DrawAll();
  void PrintAll();
private:
  TrackerHitsCollection* TkHitCollection;
  int verboseLevel;
  std::map<int,int> HitMap;
  int detId;
  int trackID;

  static const int tkIdOffset = 12;
  static const int detIdMask = 0xFFF;
  static const int tkIdMask = 0xFFFFFFFF;


};
```

```cpp
#endif
```