```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.         *
// *******************************************************************
//
/// \file electromagnetic/TestEm3/include/HEPDSWDetectorConstruction.hh
/// \brief Definition of the HEPDSWDetectorConstruction class
//
// $Id$
//
//....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......
//....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......

#ifndef HEPDSWDetectorConstruction_h
#define HEPDSWDetectorConstruction_h 1

#include "G4VUserDetectorConstruction.hh"
#include "globals.hh"

#include "SatelliteConstruction.hh"
#include "HEPDBoxConstruction.hh"
#include "CalorimeterConstruction.hh"
#include "TrackerConstruction.hh"


class G4Box;
class G4LogicalVolume;
class G4VPhysicalVolume;
class HEPDSWMaterial;
class HEPDSWDetectorMessenger;


//....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......

class HEPDSWDetectorConstruction : public G4VUserDetectorConstruction
{
public:

  HEPDSWDetectorConstruction();
 ~HEPDSWDetectorConstruction();

  G4VPhysicalVolume* Construct();

  void SetWorldDimensions(G4double aX,G4double aY,G4double aZ);

  inline void SetCalorimeterDetector(G4bool aDet){useCalorimeter=aDet;}
  inline void SetTrackerDetector(G4bool aDet){useTracker=aDet;}
  inline void SetSatelliteDetector(G4bool aDet){useSatellite=aDet;}
  inline void SetHEPDBoxDetector(G4bool aDet){useHEPDBox=aDet;}
```

```cpp
  inline void SetSatelliteConfiguration(G4String aConfig){theSatelliteConfig=aCo
nfig;}
  inline void SetHEPDBoxConfiguration(G4String aConfig){theHEPDBoxConfig=aConfig
;}
  inline void SetCaloConfiguration(G4String aConfig){theCaloConfig=aConfig;}
  inline void SetTrackerConfiguration(G4String aConfig){theTrackerConfig=aConfig
;}

  inline G4double GetWorldSizeX(){return fworldHalfX*2;};
  inline G4double GetWorldSizeY(){return fworldHalfY*2;};
  inline G4double GetWorldSizeZ(){return fworldHalfZ*2;};

  void CaloSetCaloMaterial(G4String aMat1,G4String aMat2);
  void CaloSetVetoMaterial(G4String aMat);
  void CaloSetPoronMaterial(G4String aMat);
  void CaloSetCarbonFiberMaterial(G4String aMat);
  void CaloSetHoneyCombMaterial(G4String aMat);
  void CaloSetNumberOfCrystalLayer(G4int aVal);

  void TrackerSetSiliconMaterial(G4String aMat);
  void TrackerSetKaptonMaterial(G4String aMat);
  void TrackerSetCarbonFiberMaterial(G4String aMat);

  void HEPDBoxSetBlanketMaterial(G4String aMat1,G4String aMat2,G4String aMat3,G4
String aMat4);
  void HEPDBoxSetBlanketMaterial(G4String aMat);
  void HEPDBoxSetWallMaterial(G4String aMat);
  void HEPDBoxSetWallMaterial(G4String aMat1,G4String aMat2);

  void SatelliteSetBlanketMaterial(G4String aMat);
  void SatelliteSetWallMaterial(G4String aMat);

  const G4VPhysicalVolume* GetWorld()          {return fPhysiWorld;};

private:

  G4bool          useSatellite;
  G4bool          useHEPDBox;
  G4bool          useCalorimeter;
  G4bool          useTracker;

  HEPDSWMaterial*     pMaterial;

  G4double          fworldHalfZ;
  G4double          fworldHalfY;
  G4double          fworldHalfX;

  G4Box*          fSolidWorld;
  G4LogicalVolume*  fLogicWorld;
  G4VPhysicalVolume* fPhysiWorld;

  SatelliteConstruction*    fSatelliteBuilder;
  HEPDBoxConstruction*      fHEPDBoxBuilder;
  CalorimeterConstruction*  fCaloBuilder;
  TrackerConstruction*      fTrackerBuilder;

  G4String theSatelliteConfig;
  G4String theHEPDBoxConfig;
  G4String theCaloConfig;
  G4String theTrackerConfig;

  HEPDSWDetectorMessenger* fDetectorMessenger;
};

//....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......

#endif
```