```
/*
 * ParticleGrid.h
 *
 *  Created on: Nov 23, 2013
 *      Author: arjunnar
 */

#ifndef PARTICLEGRID_H_
#define PARTICLEGRID_H_
#include <vector>
#include <vecmath.h>
#include "tuple.h"
#include  <map>
#include <list>

typedef std::vector<std::vector<std::vector<std::vector<int> > > > Grid3D;

using namespace std;

class ParticleGrid
{

public:
    ParticleGrid();
    ParticleGrid(Vector3f origin, float sizeX, float sizeY, float sizeZ);
    virtual ~ParticleGrid();
    std::vector<int> getNeighborParticleIndexes(int ParticleIndex, Vector3f
&particleLoc);
    void initializeGrid(std::vector<Vector3f> &particleLocations);

    // Made this public for testing purposes only
    Tuple::tuple<int, 3> getGridCoordinates(Vector3f &particleLoc);

    float getSideLengthX()
    {
        return sideLengthX;
    }

    float getSideLengthY()
    {
        return sideLengthY;
    }

    float getSideLengthZ()
    {
        return sideLengthZ;
    }

private:
    // Instance variables
    Vector3f origin;
    Vector3f topRightCorner;
    static const int NUM_CELLS_PER_DIMEN = 100;
    std::vector<std::list<int>> grid;

    float sideLengthX;
    float sideLengthY;
    float sideLengthZ;

    float gridSideLengthX;
    float gridSideLengthY;
    float gridSideLengthZ;

    map<int, Tuple::tuple<int, 3>> mapIndexesToGridCoords;

    // Helper functions
    inline bool isCoordValid(int val);
    inline int getGridIndex(int i, int j, int k);
    inline std::list<int> getGridListAt(int i, int j, int k);
};

#endif /* PARTICLEGRID_H_ */
```