```
class Field {
private:
    Grid grid;   // the grid of the field
    Solution solution;    // the solution of the field
    float min_bound, max_bound;    // the extent of field in terms of min/max bounds of
physical positions

public:
    Grid GetGrid() { return grid; }
    Solution GetSolution() { return solution; }
    float GetMinBound() { return min_bound; }
    float GetMaxBound() { return max_bound; }
    float GetScalar(float x, float y, float z); // the scalar of a given physical
position
    float GetGradient(float x, float y, float z);    // the gradient of a given
physical position
};



class Grid {
private:
    enum TYPE { CARTESIAN };  // The type of the grid (regular Cartesian for now, but
may be extended in the future)
    TYPE type;
    int dimension; // dimensions of the grid
    int num_cells;    // number of cells in the grid
    int num_points;   // number of grid points in the grid

public:
    TYPE GetType() { return type; }
    int GetDim() { return dimension; }
    int GetNumCells() { return num_cells; };
    int GetNumPoints() { return num_points; }
    int GetCellId(float x, float y, float z, vector<float>& interpolation_weights);
    // Given a physical position, return the cell that contains the position (cell id)
    void GetCornorPoints(int cell_id, vector<float>& cornor_positions);    // Given a
cell, what are the corner positions of the cell
    void GetNeighboringCells(int cell_id, vector<int>& neighboring_cell_ids);  //
    Given a cell, what are all the neighboring cells?
    bool IsInside(float x, float y, float z);   // is a point inside the grid or not
};
```

```
class Solution {
private:
    vector<float> scalars;
    vector<float> gradients;

public:
    void SetValue(float x, float y, float z, float value);   // Set a scalar value at a given array location
    float GetValue(float x, float y, float z);  // Get a scalar value at a given array location
    void OutputArrayToFile(); // Output the array to a file
    void InputDataToArray();  // input the data into your internal array
};
```