Gebze Institute of Technology
Department of Computer Engineering
CSE 241/501
Object Oriented Programming
Fall 2014
Homework # 5
Due date Nov 7th 2014

In this homework, you will modify your **`GameOfLife`** and **`Cell`** classes of HW4 to make them work with operator overloading.

For the **`Cell`** class, overload the following operators

- operators <, >, >= , >=, == and != for comparing two **`Cell`**s. One **`Cell`** object is smaller than the other if the Y components is smaller. If Y components are equal, then check the X component.
- Operators ++ and - - that increment and decrement the X and Y components by one. Overload both prefix and postfix operators.
- Stream insertion and extraction operators

For the **`GameOfLife`** class, you will overload the following operators

- Operator++ (both posfix and prefix) will advance the game by one step. It will return the expected results.
- Operator- - (both posfix and prefix) will undo the game by one step. It will return the expected results. You can undo all the game moves back to the beginning of the game. (Hint: you will need to think about this operator a little).
- Operator+ will take one **`GameOfLife`** object and one **`Cell`** object. It will return a new **`GameOfLife`** object that will include the passed **`Cell`** object.
- Operator- will take one **`GameOfLife`** object and one **`Cell`** object. It will return a new **`GameOfLife`** object that does not include the passed **`Cell`** object.
- Operator+ will take two **`GameOfLife`** objects and it will merge these two games and will return the merged **`GameOfLife`** object. The parameter objects are not changed.
- Operator- will take two **`GameOfLife`** objects and it will return a new **`GameOfLife`** object that includes all the **`Cell`**s from the first object except the **`Cell`**s of the second object.
- Overload the [] operator such that if g is a **`GameOfLife`** object, g[10][5] will return the **`Cell`** at row 10 and column 5. If there is no such **`Cell`**s, then it will return a new **`Cell`** with position (-1000,-1000). (Hint, g[10] returns a vector of **`Cell`** objects for the 10th row)
- Operator() with two parameters behaves exactly the same as g[10][5].
- Operator+= that takes another object **`GameOfLife`** as parameter and merges the living **`Cell`**s of the other game into this game.
- Stream insertion operator that prints the game on the screen

Write your main function to test new classes. Make at least 5 objects of class **`GameOfLive`** and show the results of each operator overload.