ASSIGNMENT 3
CELL SIMULATOR (V1.01)

Specification

For this assignment, write a program that simulates the growth and death of a population of cells. Although there are many ways to implement this simulator, here, you must use two classes (`class Board` and `class Cell`) for Part A and a third class (`class Cancer`) for Part B. Completion of Part A will allow you to attain a maximum of 80 marks while completion of Part B will allow you to attain the full 100 marks. You already have the material necessary to complete Part A but will need Lecture 7 to complete Part B.

Part A – Normal Cells

1. Main menu. At the start of the program, ask the user whether he or she would like to simulate normal cells or cancer cells. If the selection is valid, then ask the user for the confluency percentage.

2. Generate board. Generate a board with 20 rows and 75 columns that represent the location of cells. Each location should be represented by a '`O`' for a normal cell, '`X`' for a cancer cell, and ' ' (space) for no cell (or dead cell). Based on options selected in the main menu, 'randomly' distribute cells onto this board by first setting the seed using `srand(1)` and then using the `rand()` function to generate 'random' numbers. These functions are from `<cstdlib>`. For example, if normal cells are selected with a confluency of 10, then fill 10% of the board with '`O`' and 90% of the board with a space (' ').

3. Simulate cell death and birth. The state of the board should be updated every time the user presses 'enter'. With each update, increment a time index, starting from 0. The board should be updated using the following rules:

Death. If a cell is alive, it will die under the following circumstances:

- Overpopulation: If the cell has four or more alive neighbours, it dies.
- Loneliness: If the cell has one or fewer alive neighbours, it dies.

Birth. If a cell is dead, it will come to life if it has exactly three alive neighbours.

Stasis. In all other cases, the cell state does not change.

Display the time index and the total number of cells on the board. Finally, use `system("cls")` to clear the console screen before every update.

4. Exiting the simulator. Quit the program when the user enters '`q`'.

Part B – Cancer Cells

Simulate cancer cells that have a slight resistance to death from overpopulation. Use the same rules as normal cells, but replace the overpopulation rule with the following.

- Overpopulation: If the cell has *five* or more alive neighbours, it dies.

The `class Cancer` must be derived from `class Cell` and use `virtual` functions.

Implementation rules

Your `class`es must have a complete set of constructors, destructors, and access functions. The `Board` class should be the main user class. In other words, the `main()` function will never need to use `Cell` or `Cancer` classes directly. `Board` should call `Cell` and `Cancer` as needed.

Submission Procedure

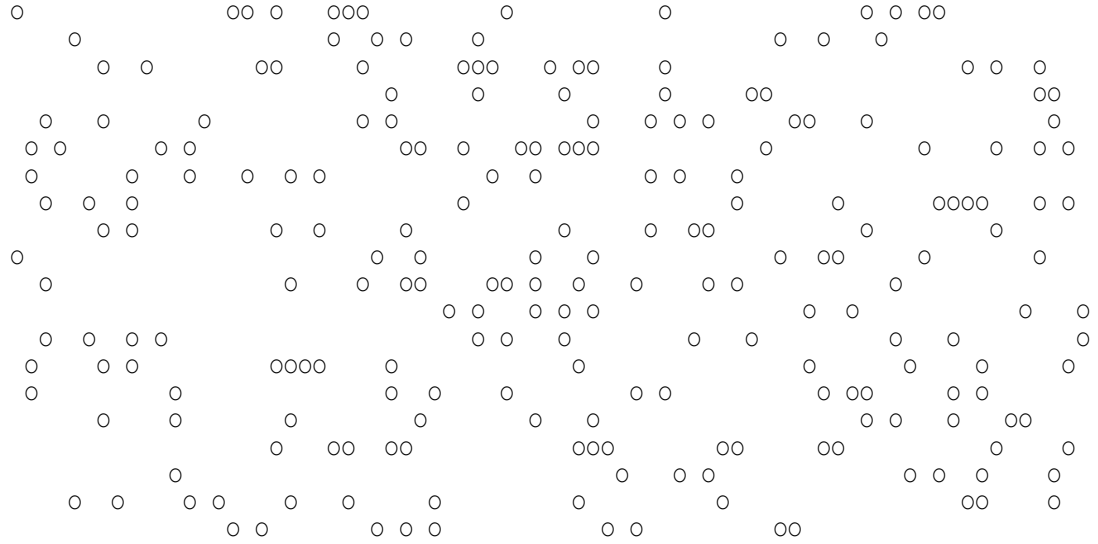
Submit only one file labelled '`cell_sim.cpp`' through the "Assignment 3 – Submission Portal" on blackboard (i.e., do not submit the `.fa` files). **The submission deadline is 4 PM on 12 December 2016**. Submissions are NOT possible after this deadline.

Example class interface

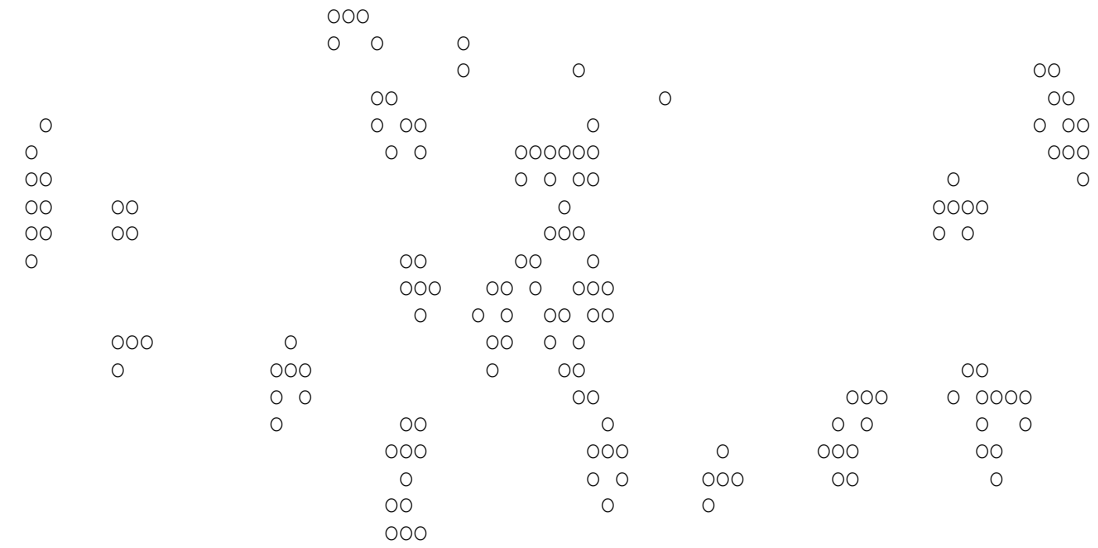
```
int main()
{
	int selection = 0;
	int confluence = 0;
	int c;
	Board board(20,75);
	string trash;
	system("cls");
	cout << "Welcome to the cell simulator" << endl;
	cout << endl;
	cout << "Select your cell type: (1) normal cells or (2) cancer cells" << endl;
	while ( (selection < 1) || (selection > 2))
	{
		cout << ">";
		cin >> selection;
	}
	getline(cin,trash);
	cout << "Select the confluence percentage (%)" << endl;
	while ( (confluence <= 0) || (confluence > 100))
	{
		cout << ">";
		cin >> confluence;
	}
	getline(cin,trash);

	board.seed_cells(selection, confluence);
	system("cls");
	while(c!='q')
	{
		system("cls");
		cout << "time:            " << board.get_time() << endl;
		cout << "number of cells: " << board.get_num_cells() << endl;
		board.display();
		c = cin.get();
		board.next_state();
	}
	return 0;
}
```

Example Program Output 1

```
Welcome to the cell simulator

Select your cell type: (1) normal cells or (2) cancer cells
>1
Select the confluence percentage (%)
>15

time:            0
number of cells: 231
```



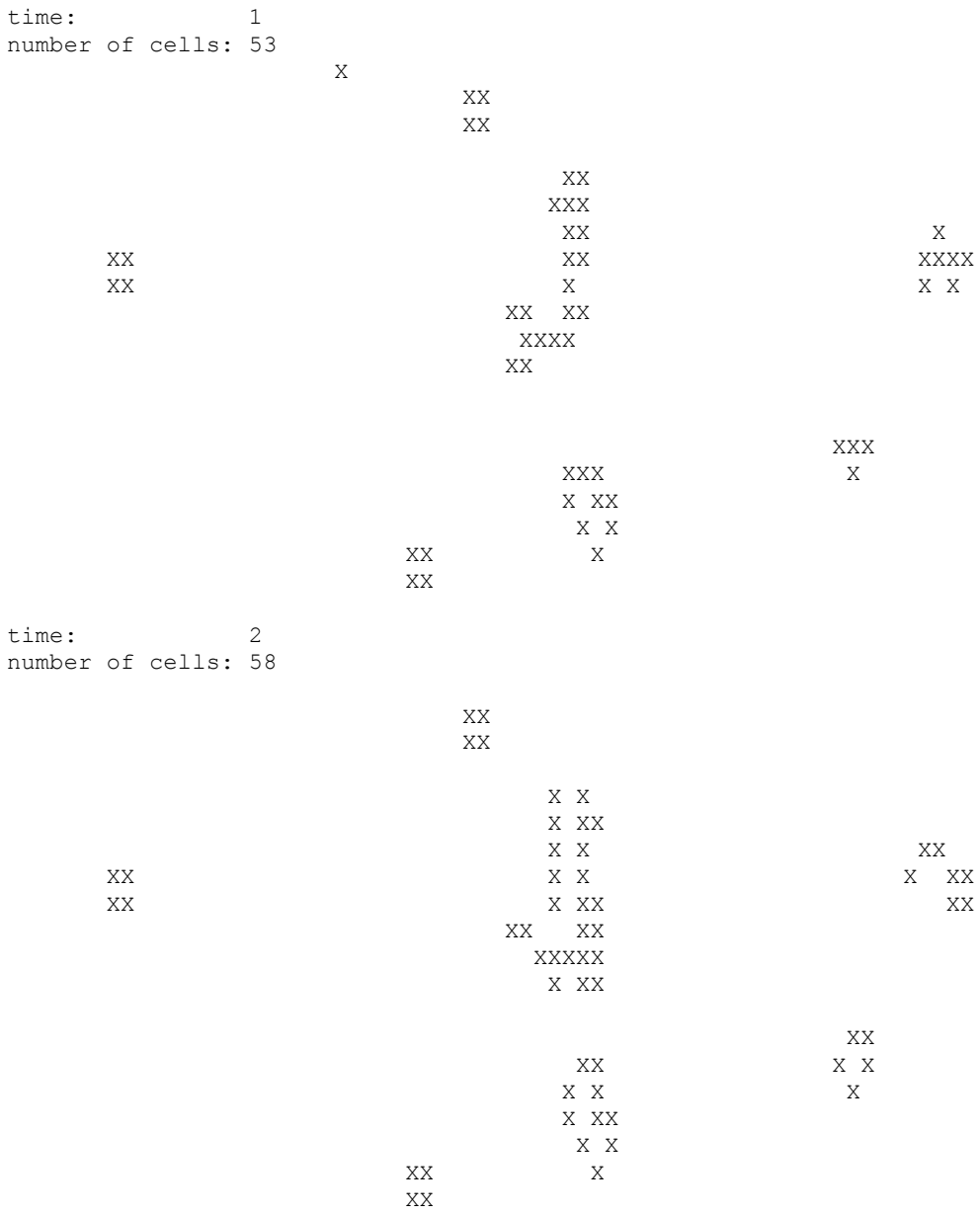
```
time:            1
number of cells: 148
```



```
time:            2
number of cells: 124
                     OOO
                     O OO
                       O O                                                OOO
                       O O                                                  O
                         O          O O O                                 O O
 OO                                 O    OO                               O O
O                                      OOO                      O OO      OO
O O    OO                                O                      O   O
O O    OO                                OOO
 O                           O      O O
                             O      O O O OO
        OO                           O O O  O
       O O         O               OOOO  OO
                  O                O OOO                      OO     OO OO
                  OOO                 OOOOO                    O         OO
                  O         O O        O                  O OO       OO OO
                            O O           O       O       O  O        O
                                          O      O O       OO
                            O                    O
                            OO
q
```

Example Program Output 2

```
Welcome to the cell simulator

Select your cell type: (1) normal cells or (2) cancer cells
>2
Select the confluence percentage (%)
>10

time:            0
number of cells: 160
X              XX      XX          X                        X X XX
                      X          X                        X
      X  X                       XX     X     X                    X X  X
                           X           X            XX                    X
  X                       X                  X X       X    X
   X                         XX  X      XXX           X          X       X X
 X          X   X  X               X       X     X
  X  X  X                         X                                XXXX     X
      X X         X  X     X          X     X  XX          X        X
                           X          X   X            X  XX     X       X
  X                       X  XX     X X  X   X    X
                               X X   X   X             X
  X  X                             X   X        X   X         X   X        X
 X    X X         X       X            X                       X          X
 X         X              X               X              XX     X X
      X    X       X        X       X   X                    X X       XX
                        XX   X          XXX        X      X              X
                                           X   X X             X X  X    X
    X  X    X          X     X         X        X              X     X
                X       X X X           X X         X
```


time: 1
number of cells: 53
time: 2
number of cells: 58