Gebze Institute of Technology
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2014
Homework # 6

MURAT ALTUNTAŞ
111044043

```
class Cell
{
public:
        /* No parameter constructor */
        Cell();
        /* 2 Parameter Constructor */
        Cell(int koorX, int koorY);
        /* getters */
        int getX() const {return coordinateX;}
        int getY() const {return coordinateY;}
        /* setters */
        void setX(const int);
        void setY(const int);
        /* logical operators */
        bool operator<(const Cell& other) const;
        bool operator>(const Cell& other) const;
        bool operator<=(const Cell& other) const;
        bool operator>=(const Cell& other) const;
        bool operator==(const Cell& other) const;
        bool operator!=(const Cell& other) const;
        // post increment/decrement operators
        Cell operator++(const int ignore);
        Cell operator--(const int ignore);
        // pre increment/decrement operators
        Cell& operator++();
        Cell& operator--();
        /* i/ostream functions */
        friend std::ostream& operator<<(std::ostream& outs, const Cell& other);
        friend std::istream& operator>>(std::istream& ins, Cell& other);

public:
        int coordinateX;
        int coordinateY;
};

class GameOfLife
{
public:
        /* No parameter constructor */
        GameOfLife();
        /* copy constructor */
        GameOfLife(const GameOfLife& other);
        /* getters */
        int getRow() const {return row;}
        int getColumn() const {return column;}
        int getCapacity() const {return capacity;}
        int getUsed() const {return used;}
        int getIUsed() const {return indexUsed;}
```

```
		/* setters */
		void setRow(const int);
		void setColumn(const int);
		void setCapacity(const int);
		void setUsed(const int);
		void setIndexUsed(const int);
		/* Dosyadan okuma fonksiyonu */
		bool readFile(const char* fileName);
		/* Dosyaya yazma fonksiyonu */
		void writeFile(const char* fileName);
		/* board i ekrana basan fonksiyon */
		void printBoard();
		/* Hucrelerin yasama yada olme durumunu control eder
		   bir sonraki adimi hesaplar */
		void play();
		/* bir objedeki canli hucreleri digerine ekler */
		void addLiveCell(const GameOfLife & other);
		/* oyunlardaki toplam canli hucre sayisini donduren fonksiyon */
		static int totalNumOfLiveCell();
		/* Board boyutlarini degistiren fonksiyon */
		void resizeBoard(const int rw, const int clmn);
		/* Gonderilen Objenin Row sayisini donduren Fonksiyon */
		int returnRow(const GameOfLife & other) const {return other.getRow();}
		/* Gonderilen Objenin Column sayisini donduren Fonksiyon */
		int returnColumn(const GameOfLife & other) const {return
other.getColumn();}
		/* assignment operator */
		GameOfLife& operator=(const GameOfLife& other);
		/* post increment/decrement operators */
		GameOfLife operator++(const int ignore);
		GameOfLife operator--(const int ignore);
		// pre increment/decrement operators
		GameOfLife& operator++();
		GameOfLife& operator--();
		// binary operators
		GameOfLife operator+(const Cell& other);
		GameOfLife operator-(const Cell& other);
		// index operator
		Cell* operator[](const int ind);
		/* parantez operatoru */
		Cell operator()(const int ind1, const int ind2);
		// compound assignment operators
		GameOfLife& operator+=(const GameOfLife& other);
		/* i/ostream functions */
		friend std::ostream& operator<<(std::ostream& outs, const GameOfLife&
other);
		/* destructor */
		~GameOfLife();
```

```
    private:
        int row;    /* num of row */
        int column; /* num of column*/
        int capacity; /* array capacity */
        int used;     /* array used */
        int indexUsed; /* for index */
        static int numOfLiveCell;
        Cell* canliHucreler;
        Cell* geciciHucreler;
        Cell tmpObj; /* atamalar icin yardimci obje */
        typedef Cell* cellPtr;
        cellPtr* history; /* oyunu geri almak icin tanimlanmis */
        int* usedArr; /* used larin tutuldugu array */
        int moveCounter; /* haraket sayisi */
        /* Komsu hucrelerdeki yasayan canli hucre sayisini donduren fonksiyon */
        int controlNumOfLiveCell(const int boardRow, const int boardColumn);
        /* bir board'dakini digerine aktarma fonksiyonu */
        void transfer();
    };
```

Ödevin içeriği:

Cell ve GameOfLife classlarından oluşmaktadır. Cell objeleri koordinat belirtir.
GameOfLife objeleri birer oyundur.
5. ödevden farklı olarak vektör yerine dynamic array kullanılmıştır.
Testin ekran görüntüleri screenshots dosyasının içersindedir.

Ödevin Çalıştırılma Şekli:

```
g++ -c HW06_111044043_GameOfLife.cpp HW06_111044043_Cell.cpp HW06_111044043_TEST.cpp

g++ -o hw06 HW06_111044043_GameOfLife.o HW06_111044043_Cell.o HW06_111044043_TEST.o

./hw06
```

Ya da Makefile çalıştırabilirsiniz.
(Dosyanın bulunduğu klasöre girip, Terminale **make** yazmanız yeterli olacaktır.)