Homework 2

The aim of the homework is to extend the implementation of class Ctable from homework 1 by defining operators (not only, read the text below very careffully).

First, perform the following exercises (tests):

1. Create two CTable objects as in the example below and perform the following operation (do not define any operators before)

```
CTable  c_tab_0, c_tab_1;
c_tab_0.bSetNewSize(6);
c_tab_1.bSetNewSize(4);
c_tab_0 = c_tab_1;
```

How did the above program finish? Why?

//!!! WE CANNOT ASSIGN A OBJECT COMPLETELY.
WE SHOULD USE OPERATOR OVERLOADING TO ASSIGN OBJECTS.

Becasuse Object is not pritimitive type. It is necessary to start copying from primitive type. Without using pointer(pass by reference).

2. Remove the destructor from the CTable class. Has something changed? Why?

C++ language hasn't garbage collector like java or similar high level languages. We should use destructure to delete(free) dynamic allocated memories to prevent memory leakage. This kind of mistake will be very dangerous for our program.

3. Implement the method `void vSetValueAt method (int iOffset, int iNewVal),` **which allows you to enter values into an array (if you don't already have it). Enter values e.g. 1,2,3,4 ... to c_tab_0 and 51,52,53,54 to c_tab_1. Write the** `vPrint()` **method that displaying an array on the screen and execute the program below.**

```
CTable  c_tab_0, c_tab_1;
c_tab_0.bSetNewSize(6);
c_tab_1.bSetNewSize(4);
/* initialize table */
c_tab_0 = c_tab_1;
c_tab_1.vSetValueAt(2,123);
c_tab_0.vPrint();
c_tab_1.vPrint();
```

What subtitles were displayed on the screen. Are you able now explain program behavior from exercise 1 (point 1).

prints
51 52 53 54
51 52 123 54

c_tab_1 assigned to c_tab_0 with operator= overloading. In operator overloading we assigned object's contents in pieces

Now you ready to implement the operator+ for the CTable class, which returns a concatenation of two arrays.

Mustafa Tayyip BAYRAM
257639