**Mandatory Assignment # 2**

UNIQUE: Deadline for 100%: Monday, October 23rd, 2020 at 8:00 am.

*For this HM you can work with a teammate (your choice). Maximum two per team.*

*Because you are allowed to work with a teammate and two brains think more than one, this assignment will not have extension and its closing date is as shown above.*

* Go to your textbook and read about modular code (pages 446-451)
* Your task is to implement a mini-data base for a bank, you will be allowed to use ONLY classes, **you are NOT allowed to use structures.**
* You can simulate a data-base by using vectors of objects.
* You are **NOT allowed** to use an array of objects to simulate the data-base. **(-100 if done)**

Your code will need two different classes:

Class # 1 (date)

This class must include the following three member variables: **Year**; **Month**; **Day**; and a **programmer defined default constructor without parameters**.

Class # 2 (account)

This class must include following member variables: **Account number**; Account type; Name; Balance; **Date of last payment** (this is an object of class “date” as defined by class # 1). **A programmer defined default parametrized constructor**, **and a programmer defined destructor**.

**Regarding Account types: only two, checking (c) and savings (s). A character will suffice.**

Each one of the following descriptions is a function that you are required to implement.

**MEMBER FUNCIONTS – METHODS**

1. A function that will allow the user to enter information, and associate it with an instance of class #2
2. A function that will allow the user to observe on screen the content of **ALL** the elements of our data base **that have valid information**. Information must be presented in a clear, simple, readable, secure, distinguishable, and organized manner.
3. A function that will allow the user to **increase (deposit)** the balance of a chosen bank account. The account search can be conducted using any search algorithm of your preference. I strongly advise you to choose the easiest one that you can find, do not overcomplicate things. This function MUST show that the balance really changed. Think about a bank, what are you supposed to see or present to the teller so that he/she can confirm a successful transaction?.
4. A function that will allow me to **decrease (withdraw)** the balance of a chosen bank account. The account search can be conducted using any search algorithm of your preference. I strongly advise you to choose an easy one, do not overcomplicate things. This function MUST show that the balance really changed. Think about a bank, what are you supposed to see or present to the teller so that he/she can confirm a successful transaction.
5. A function that will allow me to **transfer money between different accounts for the same client**.
   1. Savings to Checking
   2. Checking to Savings
6. A function that will allow me to **transfer money between different accounts for different clients**.

**FAQ**

* **How many (MAX) DB-Elements your data base should be able to host?:** 50.
* **For how many DB-Elements my professor will enter Info (for grading purposes)?:** as many as he wants, it can be 5, 4, 10, etc. Your code must be design in such a way that it will allow the instructor to decide when to stop entering info into the DB (this is Function 1)
* **For how long is suppose my code to be active?:** for as long as your professor consider it necessary. Your code must be design in such a way that it will allow the instructor to decide when to stop entering info into the DB (this is Function 1)

**Code Presentation:**

MUST BE AS FOLLOWS: **(-5 points if not respected)**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Assignment #2

// **Student 1**: John Doe

// **Student 2**: Jane Doe

// Data Structures Date: Date of Submission (M/D/Y)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Place your general program documentation here. It should

// be quite a few lines explaining the programs duty carefully.

// It should also indicate how to run the program and data

// input format, filenames etc

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// **FOR ALL FUNCTIONS function Name::MethodName()**

// Parameters: List them here and comment

// A discussion of what the method/function does and required

// parameters as well as return value.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**READ NEXT SECTION**

**Deliverables via D2L:**

1. Create a Folder name as Follows: TeamNameA2
2. Copy and paste all the files of your modular coded solution inside this folder
3. Zip the previous folder and name the zip file as follows **TeamNameA2.zip**

Example: if your team name is the Super Coders, then the ZIP file to be submitted MUST be named “SuperCodersA2.**zip**”. **10 points will be substracted if you do not name your file properly.**

**10 points will be substracted if your names are not included in your code.**

**IMPORTANT: Correct solutions without modular code will be considered invalid and a grade of zero assigned to the Homework.**

**Deliverable in class:**

* + a hard copy of your code **[-5 points if not delivered at the beginning of class]**

**[-10 points if the pages are not stapled together]**

UNIQUE: Deadline for 100%: Monday, October 23rd , 2020 at 8:00 am.

o=====o=====o=====o=====o=====o=====o=====o=====o=====o=====o=====o=====o=====o=====

**Rubric summary**

* if your code does not compile and executes in Visual Studio [-90 points]
* If your hardcopy is not delivered at the beginning of class [-10 points]
* If your pages are not stapled together [-10 points]
* If your file is not named as instructed [-10 points]
* If your names are not included in your code [-10 points]

Some other associated with presentation (readable, secure, etc) [-8 each]

* If your solution is not a “modular coded” solution [-100 points]
* Using Structures [-100 points]
* Using Arrays [-100 points]
* If your code is not delivered by the deadline [-100 points per day]
* Not respecting the code presentation (see previous page) [-5 points]
* Not designing your DB for 50 elements [-8 points]
* Your code does not implement the following successfully
  + **For how many DB-Elements my professor will enter Info (for grading purposes)?:**

[-20 points]

* **For how long is suppose my code to be active?:** [-8 points]
* **Incorrect implementation of a function** [-8 per function]

**All these are cumulative.**