**Project-4**

**Objective:**

**The purpose of this project is to expose you to:**

**Classes, objects, data members, member functions, and header files.**

.

**Problem Specification:**

**Design a class that will do the following:**

**Accept a student’s name (string) and maintains an array of test scores (float). If there is room you can add a score at the first available position. That is the position immediately following the last added score. If the array is empty, you add at the first position. If the array is full, then you cannot add and must indicate that. If there are scores in the array, you can remove the last score; scores in the middle or front may not be removed. If the array is empty then you cannot remove and must indicate that. You can also clear the array of all the scores, but before you do that you must print the name and contents of the array backward.**

**Requirements:**

* Create a CRC card as explained in chapter 1.
* Specify each method’s purpose, describe its parameters, and write a pseudo-code version of its header as explained in chapter 1.
* Create a class diagram showing the class name, public and private members.
* You are to create a header file that contains the class declaration and interface for the class that will maintain objects of the class.
* You are to define a default constructor that will ask for a student name to be entered from the keyboard and stores it. It also initializes a counter that keeps track of the number of scores in the array and is maintained when you add, remove, or clear.
* You have to define a null destructor.
* The maximum number of test scores is 5 and is stored in the class data as a static constant.
* The test scores may be integers, floats or doubles.
* Two methods are defined to determine if the array **isFull** or **isEmpty.**

The file “proj1.cpp” is the client file that tests the methods defined in the implementation file and declared in the header file called “proj1.h”

**Grading Criteria:**

5 points Good programming practices: Proper spacing, comments, use of variables, indentations and appearance of program.

5 points default constructor is defined and performs its task.

5 points destructor is defined and is null.

5 points CRC card is submitted correct and complete.

5 points a UML class diagram is submitted and is correct.

5 points a header file is used to define the class.

5 points a C++ implementation file contains member functions definitions.

10 points class templates are used to accommodate different types.

5 points the preprocessor directive #ifndef is properly used.

10 points methods specifications and pseudopod.

5 points modules coupling and cohesiveness.

5 points Accessor methods are constants and parameters passed are also constants

5 points the clear method calls the print function which prints the array elements last to first before clearing the array.

20 points Program solution fits specifications.

5 points test results for multiple runs.

**Submission Details:**

Hand-in the specifications, source program files, test results.

Due Date 03-30-2017