CS 202
COMPUTER SCIENCE II
SPRING 2016
Assignment #4

Due Date/Time: **03/15/2016 @ 11:59PM**
Total Points: **100**

Description
For this assignment you will design, implement, and test a class that represents a dynamic array of integers.

Specifications
Use of square brackets, [ ], is NOT allowed anywhere in your code except when allocating memory for a new dynamic array and deallocating a dynamic array. You will need to use pointers and pointer arithmetic instead.

- ✓ The `DynamicArray` class is represented by the following `private` variable members:
  - ❖ `arrayPtr`: An integer pointer storing the address of the array
  - ❖ `size`: Size of the array
  - ❖ `rangeMin`: The minimum value that that can be stored in the array
  - ❖ `rangeMax`: The maximum value that can be stored in the array

- ✓ The `DynamicArray` class has the following `public` function members:
  - ❖ Destructor

  - ❖ Default constructor:
    Initializes `size` to 1, `rangeMin` to 0 and `rangeMax` to 0.
    Declares a dynamic integer array of size `size` and assigns its address to `arrayPtr`.
    Initializes the first (and only) element of the dynamic array to 0.

  - ❖ Parameterized constructor:
    Initializes `size`, `rangeMin`, and `rangeMax` to the parameters passed.
    Declares a dynamic integer array with size `size` and assigns its address to `arrayPtr`.
    Assigns each array element a random integer between `rangeMin` and `rangeMax`.

  - ❖ `... Display(...)`
    Displays all the data of the object.

  - ❖ `... Minimum(...)`
    Finds the minimum integer and its frequency in the object's array.

- `... Maximum(...)`
  Finds the maximum integer and its frequency in the object's array.

- `... SwapElements(int a, int b)`
  Swaps the a^{th} and the b^{th} elements of the object's array.

- `... SubArray(int a, int b)`
  Reduces the object's array to only include the elements from the a^{th} element through the b^{th} element. This is done by:
  - Creating a new dynamic array that is only big enough to store the elements from a^{th} through b^{th}
  - Copying a^{th} through b^{th} elements from the object's array into the new array
  - Replacing the objects' array with the new array.

  Hint: Also consider whether the other member variables of the object should be updated as well.

- `... Concatenate(...)`
  Takes a `DynamicArray` object as an argument and appends its array to the end of the object's array. This will be done by:
  - Creating a new dynamic array that is only big enough to store the elements of both object's arrays
  - Copying the elements from both object's arrays into the new array
  - Replacing the objects' array with the new array.

  Hint: Also consider whether the other member variables of the object should be updated as well.

All programs must compile without errors and warnings on bobby.cs.unlv.edu using the g++ compiler. Programs that don't match these criteria will be given a zero (0).

No teamwork is allowed. All programs must be your own individual work.

Coding Style and Documentation

1. All submissions must have the following comment block at the top of their main program:

```
/*
 * Name: Your name, Class, Assignment number
 * Description: a brief description of the program.
 * Input: expected input to the program.
 * Output: expected output of the program. */
```

2. All functions and classes must have the following required documentation immediately above the function/class definition:

```
/*
 * function_identifier: brief description of what the function does.
 * parameters: what to pass into the function
 * return value: what the function returns, if any */

/*
 * class_identifier: brief description of the class
 * constructors: a list of constructor prototypes
 * public functions: a list of public function identifiers
 * private data members: a list of private data member identifiers
 * static variables: a list of any static variables */
```

3. All programs must employ proper indentation.

4. All programs must have reasonable comments throughout.

Submission
Submit your design document and source code files through WebCampus. You will submit the following four (4) files:

Design Document
- ✓ State the purpose and the functionality of your program.
- ✓ Include the specification of the `DynamicArray` class ADT.
- ✓ Include a UML diagram for the `DynamicArray` class.

DynamicArray.h:
Specification file for the `DynamicArray` class

DynamicArray.cpp:
Implementation file for the `DynamicArray` class

TestDynamicArray.cpp
The client program that tests and displays the functionality of the `DynamicArray` class.