**CS A250 – Programming Exam 1: Doubly-linked Lists**

|  |
| --- |
| **SAVE FREQUENTLY** |

**How to turn in your exam:**

1. Write the **name header** in the **TArray.h, and TArray.cpp** files.
2. Create a **folder** named **LastName\_FirstName\_A250\_PE3 \_MWAM**
3. Copy and paste the **TArray.h, and TArray.cpp** file into the newly created folder.
4. **Zip the folder**
5. **Submit**

|  |
| --- |
| **IMPORTANT -** Please note the following:   * Turn in a "**clean**" implementation; if your code is messy (poor indentation, unnecessary spacing, etc.) and does not conform to **readability** standards, you will lose points. * Do **NOT** add additional member variables to the class. * Write your code where indicated *without* moving items around. * Points for this exam also include **following** **instructions** and **paying attention to detail**.   **Other details:**   * Do **not** use a **return** statement in a **void** function. * Do **not** use the **break** and **continue** statements. * **No** need to write any comments. * The exam is worth **35 pts:**   + 1 pt. for the name header   + 3 pts. for readability |

The project contains the following files:

* **Main.cpp**
  + To **test** your functions. You will not be turning in this file.

Add the following:

* **Create a Templated Class called TArray. (use separate compilation and name the files TArray.h and TArray.cpp)**
  + The class has three private Member Variables.
    - **A** **pointer to an array of template type elements on the heap**
    - **An integer called capacity the maximum capacity of the array**
    - **An integer called numElem the number of elements in the array**
  + **Include the** **Default Constructor**
    - Initializes the array to the default length of 10
    - Initializes the Tarray to be empty
  + **Include an Overloaded Constructor that has one formal parameter**
    - The parameter is an integer used to set the length of the array
    - Initializes the Tarray to be empty
  + **Include the Big Three**
  + **Define a member function called insert**
    - Inserts elements in the array at the back unless the element is a duplicate of a value already in the array.
    - If the value is a duplicate it prints an error message “5 was a duplicate” if the duplicate value was a 5
  + **Overload the subscript operator as a member function**
    - Used to access the array using indexes
    - Prints an error message, “Not a valid index” if the index is out of range and calls the function exit();
  + **Define a member function called printArray**
    - Prints the elements in the array
    - Output would look like [1, 4, 0, -1] if the elements are integers
  + **Extra Credit (4 points) Overload the insertion operator to do the same thing as print**