Assignment 1: Templated Priority Queue

Course: IGME 309 – Real Time Simulations for Games II

Golisano College of Computing and Information Sciences

School of Interactive Games and Media

Rochester Institute of Technology

Due: Check in MyCourses

Deliverable: MyQueue.h file (only)

## Getting Started:

Video description: <https://www.youtube.com/watch?v=N0py5UvWPOo>

## Objective

Develop a templated priority queue in C++ that stores any data type supporting comparison operations. The queue should support automatic sorting by a defined metric and function as a FIFO structure with additional priority-based behavior.

## Background

In C++, the standard library includes std::queue and std::deque. While std::queue is a strict FIFO structure with limited access, std::deque supports both FIFO and LIFO behaviors with random access.  
  
This assignment asks you to build a templated queue class called MyQueue that behaves like a FIFO queue and implements priority-based sorting during insertions. This priority queue will allow data to be stored and retrieved based on a defined order using standard comparison operators (<, >, etc.).

## Requirements

You are to implement a templated class called MyQueue that includes:

### Core Functionalities

* Rule of Three:  
  - Default Constructor  
  - Copy Constructor  
  - Copy Assignment Operator  
  - Destructor
* Push() – Insert element; auto-sort the queue.
* Pop() – Remove the highest-priority (front) element.
* Print() – Output all elements (if printable).
* GetSize() – Return the number of elements.
* IsEmpty() – Return whether the queue is empty.

### Sorting

Automatically maintain order during insertions based on a comparison operator (< is expected). Sorting algorithm is your choice but must be implemented manually (no STL sorting functions allowed).

### Design Notes

* Must be implemented entirely in a header file (MyQueue.h).
* No .cpp file should be submitted.
* A provided main.cpp will be used for testing. You must not modify it.
* Your class should gracefully handle any data type that supports the required operators (e.g., <, >).
* You may assume that test objects (e.g., class Alberto) will implement the necessary operators.

## Grading Rubric (100%)

|  |  |
| --- | --- |
| Component | Weight |
| Rule of Three | 35% |
| • Constructor | 10% |
| • Copy Constructor | 10% |
| • Copy Assignment Operator | 10% |
| • Destructor | 5% |
| Push Method | 15% |
| Pop Method | 10% |
| Print Method | 10% |
| GetSize Method | 5% |
| IsEmpty Method | 5% |
| Sorting Implementation | 20% |

## Submission Instructions

* Submit only the MyQueue.h file.
* Optionally, include a readme.txt to explain your design choices or cite sources.
* Submit your file via MyCourses: Dropbox – A01: Templated Priority Queue
* Do not zip your submission.
* Unit Test: There is a provided unit test your assignment will be graded against, it is a good idea to try it on your end. You will find it under the Unit Test folder

## Academic Integrity & Code of Conduct

* You must implement this assignment from scratch.
* Do not use STL containers such as std::vector or std::priority\_queue.
* AI tools may be used for debugging and research only. You must cite your prompts and sources in a readme.txt.
* Plagiarism or copying others' code will result in a zero.

## Penalties

* -100%: Class is not named MyQueue. Or methods names are incompatible with the unit test
* -10% to -20%: Memory leaks (severity-dependent).
* -10%: Missing or unclear comments.
* -10%: Poor naming conventions (e.g., var, foo, thing, etc.). Be descriptive and consistent.