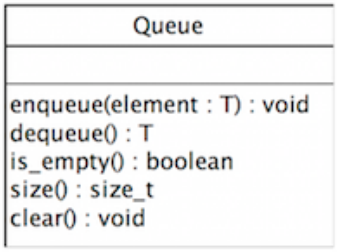
CS 36000 / ECE 49500
Programming Assignment
System Composition

Overview

In the previous assignment, you designed and implemented a standard array class. In this assignment, you will use the Array class to implement three more ADTs:

- **Stack** — an ADT that contains a list of elements such that the list has last-in, first-out (LIFO) semantics, *i.e.*, elements are inserted into the front of the list and removed from the front of the list
- **Queue** — an ADT that contains a list of elements such that the list has first-in, first-out (FIFO) semantics, i.e., elements are inserted into the of the list and removed from the front of the list; and
- **Fixed Array** — an ADT that is an array, but cannot grow or shrink.

In addition, you will implement each of the ADTs listed above using C++ templates. This will allow each ADT to work with arbitrary data types. When you download the assignment files (see below), you will notice that there are no shell files for Queue. This is because it is your job to define and implement the Queue class in C++. When defining the Queue, please make sure you follow the design in the figure below, and implement the default constructor, copy constructor, destructor, and assignment operator although they are not highlighted in the figure.



More specifically, you must implement each method in the above figure using the following specification:

- **enqueue** — adds the element to the end of the list;
- **dequeue** — removes the element at the front of the list.If there are not elements in the queue, this method throws empty_exception, similar to the stack;
- **is_empty** — test if the queue is empty. If the queue is empty, then this method returns true. Otherwise, this method returns false;
- **size** — returns the number of elements in the queue; and
- **clear** — removes all the elements in the queue.

Exception Handling. Your ADTs are expected to offer basic exception safety.

NOTICE. Use of any Standard Template Library (STL) or Standard C Library routines to implement the Array class is not permitted.

Assignment Files

The files for the assignment can downloaded from Canvas.

Development Process

For this assignment, all development must take place on the **master** branch. It is strongly recommended that you commit and push often!

Submission

All assignments must be submitted on IU Github (github.iu.edu). The name of your IU Github repository must be **cs36300-fall2018-composition**.

Please see the Canvas page titled “Repository Requirements” for the required configuration for your repository.

All source, project, and Valgrind files must be at the top-level directory of the repository. Any additional files and directories can be added to the Github repository, but they will not be reviewed or graded.

IMPORTANT. Make sure the your filenames have the correct capitalization. Failure to do so can result in our automated system not compiling and testing your code, and you potentially receiving a 0 for the assignment.