Derek Rodriguez, Derek Caprio
COP 4530, Project 2
Report and UML Diagrams

Report:

This project implements a stack and a single-ended queue data structure using dynamic arrays. The menu program contains an insert option which calls enqueue or push for queue or stack respectively and a delete option which calls dequeue or pop for queue or stack respectively as well as other common menu items such as display and clear in addition to options to check the current size of the dynamic array and the number of elements it currently stores.

Both data structures use a dynamic array for their implementations, They also both resize in the same way: if an item is inserted into an array that is full, the array size is doubled; and if after deleting an item, the array is ¼ full, then the array size is halved. In order to resize the data structure, a new array is built that is of the required size, then each element is copied into the new array, after this, the old array is deleted. Due to this resizing, the insertion and delete (push, enqueue, pop, and dequeue) were the most involved. Functions such as empty, size, and capacity were easily implemented with a couple lines of code.

Our group method of wording on this project was that Derek R. wrote the bulk of the queue header file and Derek C. worked on the bulk of the stack header file. After this, debugging and testing was shared equally with both of us working on both classes. We spent between 5 and 10 hours each on the project.

[UML diagram on next page]

UML Diagrams: