# CO222: Programming Methodology: Project 2 – Specification

Deadline: May 4th 2016 @ 11.55PM
Marks: 10%

This project is to implement a data structures library (https://en.wikipedia.org/wiki/Library_(computing)). A library (API) is a set of compiled functions that can be used in a separate program. For this project, the interface for the library is given and you will have to do 2 implementations using arrays and linked lists.

**What is a library?**

A library is a set of functions that is accessible by other programs. In C, you have used c standard input/output library by including `stdio.h`. After including it, you can use internal functions like scanf(), printf(), etc. Similarly, you should create a set of functions that can be accessed by others. Since the libraries do not have a running program on their own, the library will not have a main method.

**Library Specification**

The library should be capable of handling following data structures with integers as their data values.

- Stack – A stack is a Last-In-First-Out (LIFO) data structure. You may refer to the following link or any other suitable material for more details and the applications of a stack.
  https://en.wikipedia.org/wiki/Stack_(abstract_data_type)

- Queue – A queue is a First-In-First-Out (FIFO) data structure. You may refer to the following link or any other suitable material for more details and the applications of a stack.
  https://en.wikipedia.org/wiki/Queue_(abstract_data_type)

Each data structure should have the functions to

- Add elements (Push / Enqueue )
- Remove elements (Pop / Dequeue)
- View next element to be removed (Peek)
- Create the data structure
- Delete the data structure (Free all allocated memory)

Refer to the header file given (`StackAndQueue.h`) for more details.

Creating and deleting the data structures should be done properly in the library. Errors should be handled by returning negative values.

The library should be implemented twice. One with an array based implementation and the other with a linked list based implementation.

The library must be accessible through the given header file and it should be usable with the given program.

**Instructions**

In your implementation, you may start by implementing the *struct*s and then implement the functions. You should include the header file given.

All code should be written in separate files so that the array implementation and the linked list implementation of the library are in 2 different files (`ArrayImpl.c` and `LinkedListImpl.c`). You should not alter the given header file given.

When compiling, since the library does not have a main function, compile it without the static linking. You may use the following gcc option to do the same for `ArrayImpl.c`.

```
gcc –c ArrayImpl.c –o ArrayImpl.o
```

Then compile the full program for the array implementation with

```
gcc sampleProgram.c ArrayImpl.o –o sampleProgram
```

You may run the program with

```
./sampleProgram
```

When you run, your program will output the execution time of a selected set of queue and stack functions. You are expected to record the execution times reported by the samplePrograms for your array and linked list implementations, compare them and comment on the same.

**Deliverables (what to submit)**

1. Submit your answer (without compile error or warning) in two files with the filenames `E13yyyArrayImpl.c` and `E13yyyLinkedListImpl.c` where yyy is your registration number.
2. A report (as a text file) on the execution times and memory usages of your array and linked list representations of the stack and the queue.

You should submit by the deadline (May 4th 2016 @ 11.55PM) and late submissions will attract a penalty of 20% per day.

You may ask questions, if you have any, about the specification in the project forum and happy coding!