# CSC 225 (Fall 2016): Algorithms and Data Structures I
# Assignment 3—programming

## Due Date

- Wed, November 09, 2016

## Objectives

- Heap data structure
- Priority queues
- Heapsort
- Random number generation

## Programming Assignment

You are to implement a java class **PQ225** to practice designing, implementing and testing priority queues with the following objectives:

### Fields

An integer array, ***heapArray***

### Constructors

PQ225()

### Methods

## *Part 1: Random Number Generation*

Design and implement a java method **ranGen()** *from scratch* that generates *uniform* integer random numbers.

| Return Type | Method Description |
|---|---|
| void | ranGen(int N, int LOW, int HIGH)<br><br>generates *N* random numbers in the range *LOW* to *HIGH* and inserts them into the ***heapArray*** |

Implement your generator using the following pseudo code:

```
Procedure ranGen(N, LOW, HIGH):
        //populate heapArray with random values
        seed ← systemClock( )
        k ← 0
        while k < N do:
                //Generate a random number rn
                do
                    rn ← rand( )
                while rn < LOW or HIGH < rn:
                heapArray[k] ← rn
                k ← k + 1
End Procedure
```

## *Part 2: Building up a heap in linear time*

| Return Type | Method Description |
|---|---|
| int | size()<br>returns size of the array in O(1) |
| void | insert(int i)<br>inserts an integer i into the heap in O(log n) time |
| int | deleteMin()<br>delete the smallest integer from the heap in *O(log n)* time |
| void | makeHeap()<br>turns the unsorted integer array, heapArray into a heap in O(n) |

## *Part 3: HeapSort in O(n log n)*

Implement In Place Heap Sort.

| Return Type | Method Description |
|---|---|
| int | heapsort()<br>sorts the integer array, heapArray, using In Place Heap Sort in O(n log n).<br>Use methods from Part 2.<br>(Nothing specific to return, you can use it for testing) |

## *Part 4: Testing of Priority Queues*

| Return Type | Method Description |
|---|---|
| void | test()<br>Tests and exercises the methods developed for the first three parts of this assignment extensively. The output generated by this class must convince the marker that the classes are implemented as specified. For example:<br>• Generate heaps of different sizes (e.g., 100, 1000, 10000).<br>• Implement a test that checks whether heapsort sorts properly.<br>• Measure the performance of heapsort, by counting comparisons for inputs of different length.<br>• Chart the performance on a graph!<br><br>Your code **must** also print out results out into a textfile called **pq_test.txt**<br>This text file must have a short description what your test cases are and along with the results.<br>You have flexibility to decide and choose how extensively you want to test your methods. |