# Computer Science II — CSci 1200
# Lab 12
# Priority Queues and Binary Heaps

## Introduction

In this lab, you will use binary heaps to implement the priority queue container, as discussed in Lectures 21 and 22. Having these notes available while you are working on this lab will make it substantially easier.

Start by downloading the files:

```
http://www.cs.rpi.edu/academics/courses/spring08/cs2/lab12/test_pq.cpp
http://www.cs.rpi.edu/academics/courses/spring08/cs2/lab12/priority_queue.h
```

Then, turn off all network connections.

The code provided in these files is straightforward. `test_pq.cpp` is a driver and test program, while `priority_queue.h` is a skeleton implementation. Please take a careful look. You will complete the implementation and add to the main program in lab. In your implementation, be careful of subtracting 1 from an unsigned int whose value is 0; it is not -1!

## Checkpoints

1. Implement and test the `push` (a.k.a. `insert`) and the `check_heap` functions. Recall that `push` depends on the `percolate_down` functionality. There are two different versions of `check_heap`, but one of them just calls the other — you only need to implement the version that works with a vector. `check_heap` determines if the vector is properly a heap, meaning that each value is less than or equal to the values of both of its children (if they exist).

2. Implement and test the `pop` (a.k.a. `delete_min`) function and the constructor that builds a valid heap from a vector of values that is in no particular order. Both of these depend on proper implementation of the `percolate_down` function.

3. Finally, write and test the non-member function that sorts a vector by constructing a priority queue first.