Week 1:

Instructor: ___________________ **Office:** ___________ **Email:** __________________
Ofc Hours:

Lec 001: **TR** 9:30am - 10:45am, 5206 Social Sciences
Lec 002: **TR** 2:30pm - 3:45pm, 132 Noland Hall
Lec 003: **MWF** 8:50am - 9:40am, 5208 Social Sciences

Waitlisted Students: please come to front of class and sign by your name on waitlist.

TopHat Join Code for my lecture: ______623271__________

Announcements and Course Policies: https://pages.cs.wisc.edu/~deppeler/cs400/
Readings and Assignments: https://canvas.wisc.edu/

There is no textbook.
Assigned readings found in Modules on Canvas
Get lecture outlines (pdfs) before lecture each week. Complete during lecture.

Homework: h0 (quiz) on Reference and Review topics
Program: p1 available next week

Read: Module 1 readings

THIS WEEK:
- Learning Outcomes
- Course Mechanics and Grading
- Lecture Participation options
- Getting Started in Linux
- Testing: Black Box Unit-testing of Priority Queue

Next Week:
- Binary Search Tree (quick review)
- Balanced Search Trees
- AVL Tree: insert and remove

Course Learning Outcomes

Grades

- Attendance and class participation (10%)
- Individual assignments (10%)
- Team assignments (10%)
- Team Project - (multiple parts - 30% total)
  - o
  - o
  - o
- Midterm Exam (20%): Thursday, October 25th, 5:00 PM – 7:00 PM
- Final Exam (20%): Saturday, December 15th, 7:45 AM – 9:45 AM

Course Participation Points

Linux

What is Linux?

What is a terminal?

How do we remotely connect to a CS Linux terminal?

What can we do once we connect to a CS Linux terminal?

Open a Terminal on your Computer

Login remotely to best-linux.cs.wisc.edu

Create a local directory named: ~/private/lecture/week1
Edit-Compile-Run a Java program

Testing

How can we test a class that implements one of our ADT or data structures?

Recall: PriorityQueueADT

Operation	Description
boolean isEmpty()	return true iff the PriorityQueue is empty
void insert(Comparable p)	add priority p to the PriorityQueue
Comparable removeMax()	remove and return the highest priority from the PriorityQueue (error if the PriorityQueue is empty)
Comparable getMax()	return the highest priority from the PriorityQueue, but do not remove it (error if the PriorityQueue is empty)

What is good, bad, ugly about this code that tests a generic implementation of a Priority queue implementation?

```
try {
  PriorityQueueADT<Integer> pq = new PQ<Integer>();
  if (!pq.isEmpty())
    S.o.pln("fail …");
  try {
    pq.removeMax();
    S.o.pln("fail …"); }
  catch (Exception e) { }
} catch (Exception e) {
  S.o.pln("Fail …");
}
```

Test program structure

What is a good structure for a Java program that will be used to test a single other class?

Recall: Treenode and Tree classes

The Tree Node Class:

```
class Treenode<T> {
  private T data;
  private ListADT<Treenode<T>> children;
  ...
```

→ Memory layout of a Treenode

(assume an ArrayList is used for the ListADT):

The Tree Class:

```
public class Tree<T> {
  private Treenode<T> root;
  private int size;

  public Tree() {
    root = null;
    size = 0;
  }
  ...
```

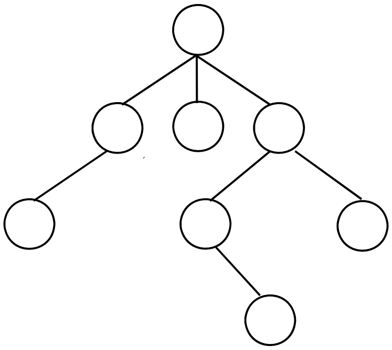
→ Memory layout
of an empty general tree:

→ Memory layout
of a general tree with a root node having 3 children:

Determining Height of a General Tree

Given that the height of a tree is the length of a path from the root to the deepest leaf.

→ Write a recursive definition for the height of a general tree.



→ Complete the recursive height method based on the recursive definition.
Assume the method is added to a Tree class having a root instance variable.

```
public int height() {
```