CPSC221 Syllabus (2017W2-2019W2)

This course teaches a variety of mechanisms for storing collections of data in support of computation, and discusses the advantages and disadvantages associated with the different methods, and it serves as students' first glimpse at algorithms that lie at the heart of Computer Science. Students learn how to implement various data-storage structures and algorithms, and they learn how to evaluate their design choices. The combined arts of design, analysis and justification are the substance of the class.

Prerequisites: CPSC121 – discrete math, and CPSC210 – software engineering.

Meeting format: 3 lecture meetings per week + 2hr lab section. (Modified as we migrate to online instruction.)

Course deliverables: 3 Programming Projects and 3 Homework Assignments, each worth 4%; 8 Lab Exercises, each worth 1%; 5 midterms, each worth 10%; final exam, worth 18%. (approx) Assignments are primarily auto-graded, with some manual evaluation of handwritten proofs.

Approximate Schedule:
Week 1 – Analysis and Correctness of Linear Algorithms
Week 2 – Linear Sorting and Linked lists, Stacks, and Queues
Week 3 – Analysis and Correctness of Recursive Algorithms
Week 4 – Recursive Sorting and Intro to Trees
Week 5 – Tree Traversals, Binary Search Trees
Week 6 – Balanced Binary Search Trees
Week 7 – B-Trees
Week 8 – Hashing
Week 9 – Priority Queues, Heaps, Heapsort
Week 10 – Disjoint Sets, Graph Definitions
Week 11 – Graph Implementation and Traversal
Week 12 – Minimum Spanning Trees, Shortest Path

Instructional Details:

1. Lectures are Socratic and constructivist—a running dialogue between instructor and students. Daily activities include: puzzles, deliberate mistakes, and stories, just to keep things hopping.
2. We employ undergraduate course staff in a ratio of approximately 1:35 students. Their role is to help students overcome barriers to mastery of the material, primarily on the coding projects.
3. We deliberately craft staff-student interactions so that they are (almost) never one-to-one. My office hours are set up to encourage peer learning. We have "solution parties" and "exam visitations" wherein groups of students receive feedback on their work.

4. We rely on educational technologies to facilitate student communication (piazza, for example), and feedback, and to collect data that helps us identify students who might be in trouble.
5. It is impossible for us to know all of the students, but we think it's important that they believe they COULD come talk to us if they needed to do so. We work hard to anticipate student needs and address them via course infrastructure. Doing this kind of filtering assures that we actually *can* be available for those in unusual circumstances.