Course information CS340 202230

- This is a blended course. Some lectures will be posted as pre-recorded videos online weekly. In addition, in-person lectures will be delivered in CL 127 every Friday 9.30am – 10.20am, starting Sep 02, 2022. It is expected that students both watch the pre-recorded videos and attend the in-person lectures. The in-person lectures on Fridays will assume that the students have already watched the pre-recorded videos that were posted prior to the in-person lecture.
- Instructor: Dr. Sandra Zilles; email `zilles@cs.uregina.ca`
- Office hours via zoom: Wed 4.00pm – 5.30pm, Wed 8.30pm – 9.30pm, Fri 2.30pm – 3.15pm, starting Sep 01, 2022 (or else by appointment; individual in-person appointments are also an option). Temporary changes to the office hour schedule will be announced in class and on UR Courses.
- All important organizational information related to this course is posted on UR Courses, where updates will be made regularly.
- Prerequisite: CS 210

Text

Required textbook: Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Fourth Edition, Addison Wesley, 2014. We will sometimes deviate slightly from the material presented in this textbook. In particular, some of the material covered in class is **not covered by the textbook**.

Grading

20% assignments (7–8 assignments posted throughout the term; the lowest-scoring assignment of these will be ignored in the final grade)
30% midterm exam (45minute exam, participation required; date Oct 14, 2022, 9.30am–10.20am)
50% final exam (grade of at least 50% in the final exam required; if less than 50% is achieved on the final exam, then the final exam grade will be the grade for the whole course; date Dec 09, 2022, 9:00am–11:00am)

Information about exam procedures for both midterm and final exam (whether in person or online) will be posted shortly.

- NOTE: Cheating or attempted cheating in the midterm exam or the final exam may result in a fail grade for the whole course, independent of other achieved results. All cheating attempts will be reported to the Associate Dean. Please also note the policies in the undergraduate calendar. Both exams are closed-book.
- The final exam is comprehensive in the sense that it may test the whole course material.

Assignments

- All solutions to programming problems on assignments must be in C++. Please make sure your code compiles with Visual C++.
- All assignments will be submitted through UR Courses.
- All assignments have to be submitted on or before the due date indicated on the assignment sheet.
- NOTE: Both the solutions to programming assignments and the solutions to non-programming assignments must represent your own original work. Discussing assignments with other students is encouraged, but it is not allowed to share or copy solutions, partial solutions, or programming code. Non-compliance will be reported to the Associate Dean and may cause a grade of 0% for the assignment, or, in repeated cases, in a fail grade for the whole course, independent of other achieved results. Please also note the policies in the undergraduate calendar.

Topic outline

1. Algorithm Analysis (trade-off between time efficiency and memory efficiency; principle of worst-case, best-case, average-case analysis; Big-O, Big-Omega, Little-o, Theta and typical complexity classes; running time calculation rules; amortized analysis and splay trees; recurrence relations and the Master Theorem)
2. Priority Queues (binary heaps, operations, and analysis; d-heaps; leftist heaps, operations, and analysis; skew heaps; binomial queues, operations, and analysis)
3. Sorting (insertion sort and analysis; shellsort and analysis; heapsort, mergesort, and analysis; quicksort; $n \log(n)$ lower bound for comparison-based sorting; external sorting)
4. Graph Algorithms (terminology on graphs, adjacency list, adjacency matrix; topological sort; unweighted shortest paths and Breadth First Search; weighted shortest paths, Dijkstra; network flow; minimum spanning trees, Prim, Kruskal, union-find; Depth First Search, Euler circuits)
5. Complexity and Unsolvability (reduction principle; unsolvable problems; P, NP, NP-complete problems, polynomial reduction)
6. Algorithm Design Techniques (greedy principle, divide-and-conquer principle, dynamic programming, backtracking, parallel algorithms)