

Algorithm Analysis and Development

Prerequisites: A grade of C⁻ or better in CPSC 100, CPSC 101, and CPSC 141; or permission of instructor.

Instructor: David Casperson; **Office:** TL 10-2080; **Phone:** 960-6672; **Departmental Administrative Assistant:** Marva Byfield (960-6490); **e-mail:** David.Casperson@unbc.ca ; **web:** <http://casper.unbc.ca/> .

Lecture times: MWF 16:30– 17:20. **Room:** 5-174. There are *no* assigned lab or tutorial times.

Text Book: [3] is *required* for this course.

References: [2] is a good JAVA resource book. [1] is difficult reading, but contains a wealth of information on list data-structures, algorithmic analysis and sorting algorithms.

Grading Scheme:

Homework:	25%	
Midterm 1:	20%	Fri, Oct 12
Midterm 2:	20%	Fri, Nov 9
Final Exam:	35%	3h in 4–14 Dec

I reserve the right to change the weight of any portion of this marking scheme. If changes are made, your grade will be calculated using the original weighting and the new weighting, and you will be given the higher of the two.

Programming Assignments: There will be approximately four medium-sized programming assignments during the semester. Familiarity with JAVA is assumed.

Syllabus: Much of the material is from [3], in particular Chapters 2–4 and 7, with other material as time permits.

Topics include:

- Algorithm analysis and asymptotic complexity (2 weeks).
- Loop variants, loop invariants, and recursive programming (1 week).
- Templates, the Collections Library, containers, iterators, and generic programming in JAVA (1 week).
- Sorting algorithms (2 weeks).
- List classes (1 week).
- List based classes: stacks, queues, and deques (1 week).
- Tree classes (1 week).
- Introduction to decidability, complexity, and complexity classes (1 week).

Times are approximate.

Cheating: First offenses result in a grade of –100% on the assignment in question and formal notification of the College Dean. Allowing someone to copy your work is cheating. The UNBC Calendar describes academic offenses and possible penalties in more detail.

References

- [1] Donald E. Knuth, *Sorting and searching*, second ed., The Art of Computer Programming, vol. 3, Addison-Wesley, 1998.
- [2] Cay S. Horstmann, *Big java: Early objects*, 5th ed., Wiley Publishing, 2012.
- [3] Mark Allen Weiss, *Data structures and algorithm analysis in Java*, third ed., Addison-Wesley, 2012, for CPSC 200—2018.