COMP 352 Data Structures and Algorithms
Summer 2022 - Course Outline

Concordia University
Engineering and Computer Science

Instructors	**Section AA:** Nora Houari (Coordinator) E-mail: houari@encs.concordia.ca Lectures: Tuesdays and Thursdays 6:30 PM - 9:00 PM – H110 SGW Office hours: Thursdays 5:00 PM - 6:00 PM Or by appointment - send an email to make an appointment Office: ER 1151, 2155 Guy St. Tel: (514)848-2424 ext. 8763
	Section AB: Khaled Jababo E-mail: jababo@encs.concordia.ca Lecture: Mondays and Wednesdays 6:30 PM - 9:00 PM – H1070 SGW Office hours: By appointment - send an email to make an appointment
	Tutorials & POD Schedule: Please see your section website for full details.

Important Note: This is an in-person registered course, all course components (lectures, tutorials, POD, midterm and final examinations) will be on site.

1. Course Description

Pre-Requisites: COMP 232 or COEN 231; COMP 249 or BCEE 231.

Topics: Abstract data types: stacks and queues, trees, priority queues, dictionaries. Data structures: arrays, linked lists, heaps, hash tables, search trees. Design and analysis of algorithms: asymptotic notation, recursive algorithms, searching and sorting, tree traversal, graph algorithms.

2. Learning Objectives

- **Knowledge base:** Demonstrate competence in fundamentals of data structures and algorithms.
- **Problem analysis:** Analyze and state model limitations and elements of uncertainty. Formulate and calculate qualitative and quantitative qualities of the problems' inputs and outputs. Estimate computational complexity. Evaluate and pick the most appropriate approach based on relevant criteria.
- **Design:**
  - Critique/evaluate many possible diverse solutions and use techniques to evaluate different solutions with sound arguments related to the problems' requirements and constraints. Demonstrate thinking outside the box to create innovative solutions.
  - Develop a system architecture adapted to the systems application context and its requirements and constraints. Development and specification of internal and external software interfaces at different modularity levels. Describe a solution that presents enough details for implementation.
  - Write code according to design. Validate implemented systems against system requirements, specifications and constraints, as well as interface specifications.

- **Use of Engineering tools:** Demonstrate appropriate operational use of tools (e.g., algorithms, abstract data types, data structures, asymptotic complexity analysis) for specific tasks in a laboratory environment.
- **One credit** represents, for an average student, a minimum of 45 hours of workload spread across the various academic activities (Source: Article 16.1.2 of the Undergraduate Calendar.) For an average student, this suggests a minimum of 135 hours of workload for a 3-credit course, including the time spent in lectures, tutorials, laboratories, examinations, and personal and teamwork.

3. Course Materials

Textbook M.T. Goodrich, R. Tamassia, Michael H. Goldwasser. Data Structures and Algorithms in Java, 6th edition. John Wiley & Sons, 2014. ISBN 978-1-118-77133-4. (Note: 5th edition is ok.)
The book is available at the bookstore or can be rented as eTextbook.
Textbook/eTextbook URL: http://ca.wiley.com/WileyCDA/WileyTitle/productCd-EHEP002900.html
From the textbook we shall study these sections: 3.1 to 3.4, 4.1 to 4.3, 5.1 to 5.6, 6.1 to 6.2, 7.1 to 7.3, 8.1 to 8.4, 9.1 to 9.4, 10.1 to 10.3, 11.1 to 11.3, 12.1 to 12.4, 14.1 to 14.7, and 15.1 to 15.2. **You also need to study chapters 1 and 2 that will not be covered in provided lectures**.

Web Page Many resources for the course (slides, assignments, example programs, ...) will be available online through Moodle; (available through the MyConcordia portal www.myconcordia.ca), or through the instructor's website. Your instructor will inform you of the exact link to access the materials.

4. Grading

Assignments	15% (5% x 3)
Midterm	30%
Final	55%

To pass the course, you must pass the assignments, midterm and the final exam. Usually, a score of 50% is required. There are no make-ups/alternates for missed exams or assignments.
There is no a priori fixed relationship between your total percentage scored and the final letter grade assigned.

- ⇨ **Assignments:** There will be 3 assignments, all of which will be available online. These assignments play a major role in your learning of the various topics covered in this course. To pass the course, you must pass the assignments' component. Assignments must be submitted by the **due date**. No assignment will be accepted after the due date. More detailed information on assignments is provided with each assignment. Please read them carefully!
- ⇨ **Midterm and Final exams**: We will have a midterm and a final exam in this course. The formats of these two exams will be similar and include multiple choice questions, questions with detailed answers, and True/False questions requiring a short justification. The final exam will be identical for all sections.

Disclaim: In the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change.

5. Schedule

Please note that the following is a tentative schedule.

Book Chapter	Topic
-	Introduction
4	Algorithm Analysis (4.1 to 4.3)
5 + 15	Recursion (5.1 to 5.6, 15.1 to 15.2)
6	Stacks and Queues (6.1 to 6.2)
3 + 7	Vectors, Lists, Iterators and Sequences (3.1 to 3.4, 7.1 to 7.3)
8	Trees (8.1 to 8.4)
9	Priority Queues and Heaps (9.1 to 9.4)
10	Maps, Dictionaries and Hash Tables (10.1 to 10.3)
11	Search Trees (11.1 to 11.3)
12	Sorting (12.1 to 12.4)
14	Graphs (14.1 to 14.7)

Important dates

Event	**A#1**	**A#2**	**Midterm**[1]	A#3	**Final**
Due date	May 20	Jun 3	Ma 23 (Section AB) Ma 24 (Section AA)	Jun 17	*[2]

6. Behavior

All individuals participating in courses are expected to be professional and constructive throughout the course, including in their communications. Concordia students are subject to the Code of Rights and Responsibilities which applies both when students are physically and virtually engaged in any University activity, including classes, seminars, meetings, etc. Students engaged in University activities must respect this Code when engaging with any members of the Concordia community, including faculty, staff, and students, whether such interactions are verbal or in writing, face to face or online/virtual. Failing to comply with the Code may result in charges and sanctions, as outlined in the Code.

7. IP

Content belonging to the instructor, or the TAs shared in online, including, but not limited to, course notes, assignments remain the intellectual property of the faculty member and the TAs. It may not be distributed, published or broadcast, in whole or in part, without the express permission of the faculty member. Students are also forbidden to use their own means of recording any elements of lecture without express permission

[1] Midterm will be scheduled during second half of the lecture, duration 75mn.
[2] Schedule during final exams period by the exam office.

of the instructor. Any unauthorized sharing of course content may constitute a breach of the Academic Code of Conduct and/or the Code of Rights and Responsibilities. As specified in the Policy on Intellectual Property, the University does not claim any ownership of or interest in any student IP. All university members retain copyright over their work.

8. Ethical Behavior

Plagiarism The most common offense under the Academic Code of Conduct is plagiarism which the Code defines as *"the presentation of the work of another person as one's own or without proper acknowledgement."*
This could be:

- material copied word for word from books, journals, internet sites, professors course notes, etc.
- material that is paraphrased but closely resembles the original source.
- the work of a fellow student, for example, an answer on a quiz, data for a lab report, a paper or assignment completed by another student.
- a solution or Java code purchased through one of the many available sources.

Plagiarism does not refer to words alone; it can also refer to copying images, graphs, tables, and ideas. Presentation is not limited to written work. It also includes oral presentations, computer assignments and artistic works. Finally, if you translate the work of another person into French or English and do not cite the source, this is also plagiarism.

In Simple Words: Do not copy, paraphrase or translate anything from anywhere without saying where you obtained it!
Source The Academic Integrity Website: http://www.concordia.ca/conduct/academic-integrity.html

9. Graduate Attributes

As part of either the Computer Science or Software Engineering program curriculum, the content of this course includes material and exercises related to the teaching and evaluation of graduate attributes. Graduate attributes are skills that have been identified by the Canadian Engineering Accreditation Board (CEAB) and the Canadian Information Processing Society (CIPS) as being central to the formation of engineers, computer scientists and information technology professionals. As such, the accreditation criteria for the Software Engineering and Computer Science programs dictate that graduate attributes are taught and evaluated as part of the courses. The following is the list of graduate attributes covered in this course, along with a description of
how these attributes are incorporated in the course:

- **A knowledge base for engineering:** *Demonstrated competence in university level mathematics, natural sciences, engineering fundamentals, and specialized engineering knowledge appropriate to the program.* Knowledge of abstract data types: stacks and queues, trees, priority queues, dictionaries. Data structures: arrays, linked lists, heaps, hash tables, search trees. Design and analysis of algorithms: asymptotic notation, recursive algorithms, searching and sorting, tree traversal, graph algorithms.
- **Problem analysis:** *Ability to use appropriate knowledge and skills to identify, analyze, and solve complex engineering problems in order to reach substantiated conclusions.* Analyze problems and determine their constraints in order to make a choice as to what data structures and algorithms to use for their implementation.
- **Design:** *Ability to design solutions for complex, open-ended engineering problems and to design systems, components or processes that meet specified needs with appropriate attention to health and safety risks, applicable standards, and economic, environmental, cultural and societal*

considerations. Use and compose appropriate data structures and algorithms to solve a variety of problems.

- **Use of engineering tools:** *Ability to create, select, apply, adapt, and extend appropriate techniques, resources, and modern engineering tools to a range of engineering activities, from simple to complex, with an understanding of the associated limitations.* Make educated choices as to what data structures and algorithms to use to solve problems following their respective strengths and constraints.

10. Accessibility and student services

Accessibility The instructor will strive to make learning experience as accessible and inclusive as possible. If you have accessibility needs that require academic accommodations, please meet with an advisor from the Access Centre for Students with Disabilities (ACSD) as soon as possible to set up an accommodation plan. http://www.concordia.ca/students/accessibility.

List of student services

1. Access Centre for Students with Disabilities
2. Student Success Centre
3. Counselling and Psychological Services
4. Concordia Library Citation and Style Guides
5. Health Services
6. Financial Aid and Awards
7. Academic Integrity
8. Dean of Students Office
9. International Students office
10. Student Hub
11. Sexual Assault Resource Centre
12. Aboriginal Student Resource Centre

Important Notes

1. **One credit** represents, for an average student, a minimum of 45 hours of workload spread across the various academic activities (Source: Article 16.1.2 of the Undergraduate Calendar.) For an average student, this suggests a minimum of 135 hours of workload for a 3-credit course, including the time spent in lectures, tutorials, laboratories, examinations, and personal and teamwork.

2. Assignments will consist of a theoretical and a programming part. Each student must independently and separately prepare and submit the theory part of the assignment. For the programming part of the assignment students might be asked to work in teams (maximum 2 students), in such case, both team members must prepare the programming part but only one solution needs to be submitted by either member.
   For the programming part a demo of about 5 minutes will take place with the marker. One (or **both** members, if groups are permitted) **must** attend the demo and be able to explain their program to the marker. Different marks may be assigned to teammates based on this demo. The schedule of the demos will be determined and announced by the markers, and students must contact the marker to reserve their time slot. **Demos are mandatory. Failure to do your demo will entail a mark of zero for that assignment regardless of your submission. Missing your reserved demo time, will similarly result in a zero mark for the assignment regardless of your submission. There will be no substitution/replacement for a missed demo time.**

3. **Criteria used in evaluation of assignments:**
   - Correctness and Testing: the program should conform to the specification given in the assignment. This includes the proper handling of special cases and error conditions and the providing of correct results. The submitted test cases take into consideration special cases and error conditions.
   - Design: the program should be constructed from coherent, independent, and decoupled functions. A function should usually access only its own parameters and local variables.
   - Style: the program should be general-purpose and well-organized.
   - Documentation and Layout: The documentation should consist of a well-annotated program and clearly formatted output. Helpful identifiers and a clear layout are part of documentation. The documentation should include the description of your design and the algorithm implemented.
   - Efficiency: The program must implement the most appropriate method.
   - Program-User Interface: The program should be easy to use.

4. **Programming assignments:** For all programming components of your assignments, you need to use Java version 8 or later. You will be using the same computing facilities and the same computer account you used in previous courses (e.g., Comp 249). If you do not have a computer account, you can obtain it from the help desk at H-960. This account will give you access to the laboratories.
   For more information on CSE Computer accounts please visit the following website: http://www.encs.concordia.ca/helpdesk/access.html.

   If you have your own computer and prefer to use it, you may do so, but be aware that your programs must compile and run with Java 8.0, or later version, at the Concordia laboratories.

Submission format: All assignment-related submissions must be adequately archived in a ZIP file using your ID(s) and last name(s) as file name. The submission itself must also contain your name(s) and student ID(s). Use your "official" name only - no abbreviations or nick names; capitalize the usual "last" name. Inappropriate submissions will be heavily penalized. Only electronic submissions will be accepted. Students will have to submit their assignments (one copy per group in case groups are allowed) using the Moodle.

For the Java programming assignments, you have to submit the complete source code **and** the compiled files, which must be executable without changes. If this is violated, you will get a zero mark for these parts of the assignments.