EECS2101: Fundamentals of Data Structures

Common Policies for Sections M, N, X, Z – Winter 2025

Subject to Changes until: Monday, January 20

Course Syllabus

• All sections (M, N, X, and Z) are coordinated.
• You are expected to familiarize yourself with the policies that are common to **all** sections and, equally importantly, those that are specific to your **enrolled** section.
• This current document contains policies and information that are common to **all** sections.
• For policies and information specific to your **enrolled section**, see the documents and/or sites referenced by the section instructor.

1 eClass Site

– A single eClass site for all sections (M, N, X, Z):

`https://eclass.yorku.ca/course/view.php?id=134610`

This eClass site is used to contain:

- announcements applicable to all sections
- instructions of programming assignments (to be submitted via *web submit*)
- resources common for all sections (e.g., example programming test questions)
- policies and information common to all sections
- links to section-specific sites/resources

2 Course Policies

1. **Programming Assignments**

You will receive **full** marks as long as successful submissions are made by the corresponding **submission deadlines**, which are enforced **strictly**.

The format of your programming assignments is **identical** to that of the subsequent programming tests. Therefore, it would be your best interest in following the **export/submission process** (for which you will be expected to complete alone during the programming tests) and submitting work that:

– complies with the instructions (so as to avoid penalties when you submit for grading in the actual programming tests); and
– represents your **true** and **best** attempt.

The rationales of this policy are that: **1)** you can **rest assured that you will not lose any marks from assignments (as long as you submit them by the deadlines)**; and **2) you can just focus on the learning by seeking help from colleagues, TAs, and your section instructor without worrying about violating the academic honesty policy**.

Please do not abuse this policy: **you are still 100% responsible for acquiring the intended understandings and skills from these assignments**. Be advised that **later scheduled (written and programming) tests, as well as the final written exam, will be based on these assignments**, so if you chose **not** to learn the materials responsibly (e.g., relying much on your colleagues, submitting incomplete work and only intending to look at solutions when they are made available), you would risk at **poor performance** in subsequent tests and the exam.

2. **Assignment/Test Deadlines**

   Stringent deadlines are imposed on all scheduled in-person **written tests**, scheduled in-person **programming tests**, and **assignments** (to be submitted remotely via the *web submit* to the EECS server).

   An in-person, written **exam** will be scheduled by the registrar office to take place during the **exam period**.

   It is your **sole** responsibility for meeting all these deadlines.

3. **Test Dates and Locations**

   For each test, it is your **sole** responsibility for ensuring that you are available to take the test (during the chosen lecture time of your **enrolled section**).

   - Programming Test 1
     - During the lecture time of your **enrolled section**, on the **week of Feb 24**
     - Location: WSC105/106/108
     - A PDF guide will be released, at least one week prior to the test, to confirm the exact day and time.
   - Written Test
     - During the lecture time of your **enrolled section**, on the **week of Mar 10**
     - Day/Time and location to be confirmed by your **section instrutor**
   - Programming Test 2
     - During the lecture time of your **enrolled section**, on the **week of Mar 24**
     - Location: WSC105/106/108
     - A PDF guide will be released, at least one week prior to the test, to confirm the exact day and time.

4. **No Team Work Allowed for Scheduled Tests**

   All **written & programming tests** are to be completed **individually**.

   When a scheduled test takes place between different sections, until all test sessions conclude, it is considered a **a violation of academic integrity** if you communicate in any way, shape, or form with others about the test(s) already given.

5. **Plagiarism**: When submitting each of your **written tests** and **programming tests**, you claim that it is **solely** your work. It is considered as **a violation of academic integrity** if you copy or share **any** parts of your work (e.g., code, notes) during **any** stage of your development. The instructor and TAs may examine all submissions, and suspicious ones will be reported *immediately* to Lassonde as *a breach of academic integrity*. **We do not tolerate academic dishonesty**.

6. **Missed Tests**

   It is your **sole** responsibility for initiating the communication by the set deadline, or a grade penalty (e.g., a zero for the test) may be applied.
   - If you missed a programming test:
     - Contact your **section instructor** by the end of the test day in order to gain the approval to write a makeup test.
       The approval is by the judgement of your **section instructor** and may **not** be assumed as automatic.
     - Once approved, assume that there will be a **makeup programming test** scheduled on the following week (during the work hours on a week day).
       This is the only makeup opportunity that can be offerred, and once its exact time is set, you are supposed to prioritize with and accommodate for it.
   - If you missed the written test:
     - Contact your **section instructor** by the end of the test day.
7. **Accommodation**
   - Contact your **section instructor** by the end of the 1st week (**Friday, Jan 10**).
   - For each programming test, you will write them in an EECS lab machie, set up with the appropriate accommodation. Therefore, please:
     - cancel bookings of both programming tests with the alternate exam centre;
     - discuss how the written test will be accommodated with your **section instructor**;
     - keep your booking of the final exam.

     You are supposed to prioritize and accommodate with the time that is set for your programming test. Please inform your **section instructor**, at the first contact, of the scheduling contraints on your end.
8. **Late Enrolment**: Students who are not yet officially registered should assume an eventual successful enrolment into the course and are responsible for: **1)** contacting the section instructor **within Week 1** for course information (e.g., lecture materials, assignments access and deadlines); and **2)** attending lectures, submitting assignments, and taking scheduled tests in time.

   No assignment deadline extensions or deferred tests due to late enrolment will be accommodated.

3 Prerequisites

- **General Prerequisites**: A cumulative grade point average (GPA) of 4.50 or better over all previously completed Major EECS courses. The GPA computation excludes all EECS courses that have a second digit 5, or are Co-Op/PEP courses.
- LE/EECS1030 3.00 or LE/EECS2030 3.00
- LE/EECS1028 3.00 or SC/MATH1028 3.00 or LE/EECS1019 3.00 or SC/MATH1019 3.00

4 Course Description

This course discusses the fundamental data structures commonly used in the design of algorithms. At the end of this course, students will know the classical data structures, and master the use of abstraction, specification and program construction using modules. Furthermore, students will be able to apply these skills effectively in the design and implementation of algorithms.

Abstract operations on data structures are specified using pre and post conditions and/or system invariants. Trade-offs between a number of different implementations of each abstract data types (ADT) are analyzed.

Each algorithm operating on data structures is proved correct using loop invariants or induction. Both formal and informal proofs are introduced though most of the reasoning is done informally.

Data structures are coded and unit tested in an object-oriented language. Selecting the appropriate ADT and a suitable implementation depending on the application is covered.

5 Course Learning Outcomes (CLOs)

Upon completion of the course, students are expected to be able to:

CLO1 Instantiate a range of standard abstract data types (ADT) as data structures.

CLO2 Implement these data structures and associated operations and check that they satisfy the properties of the ADT.

CLO3 Apply best practice software engineering principles in the design of new data structures.

CLO4 Demonstrate the ability to reason about data structures using contracts, assertions, and invariants.

CLO5 Analyse the asymptotic run times of standard operations for a broad range of common data structures.

CLO6 Select the most appropriate data structures for novel applications.

6 (Tentative) Lecture Topics

Whereas the pace will be adjusted according to the class dynamics, the following topics are planned to be covered among all sections:

- Recursion
- Asymptotic Analysis of Algorithms
- (Iterative) Sorting: Selection, Insertion
- (Recursive) Searching & Sorting: Binary Search, MergeSort, QuickSort
- Linked Lists: Singly-Linked vs. Double-Linked
- Abstract Data Types (ADTs)
- Design by Contracts (e.g., preconditions, postconditions, invariants)
- Generics in Java
- Abstract Classes and Interfaces
- Stacks, Queues, Deque
- Maps, Hash Table
- General Trees, Terminology, Tree Traversals, Binary Trees (BTs)
- Binary Search Trees (BSTs)
- Balanced BSTs, AVL Trees
- Heap, HeapSort
- ADT: Priority Queue

• Additional topics may be covered at the discretion of your section instructor.
• The depth to which each topic topic is covered may also vary depending on the instructor.
• These discrepancies will be reflected **both** on the section-specific written test **and** on the section-specific questions in the final exam.

7 Grading Scheme

		Subtotal
4 Assignments (1.25% each)	5%	30%
Programming Test 1	10%	
Programming Test 2	15%	
Written Test	15%	70%
Exam (Cumulative)	55%	

8 Final Exam: Cumulative & Substantial

- Your final exam will be ***cumulative***: it will cover **all** study materials.
  - It will be an opportunity for you to **continually** *synthesize* topics that are connected.
- Therefore, your final exam will be the **most substantial** grading component.
  - It assesses how competently you can apply the learned concepts and skills.
  - The best preparation is to constantly review and reflect on the covered topics.
- The exam to be written by all sections (M, N, X, Z) will consist of two parts:
  - a core set of questions, designed and agreed by all section instructors; and
  - another set of questions designed specifically by your section instructor.

9 Expected Weekly Workload

- Lassonde's recommendation is 3 – 4.5 hours per credit: *9 – 13.5 hours* for a 3.00 course.
- "In-Class" Hours:
  - In-Class Lectures [3 hours]

    Optional: Office Hours, Problem Solving Sessions
- "Out-of-Class" Hours:
  - Completing Assignments, Studying for Lectures/Tests [6 to 10.5 hours]
- Given that this is a ***foundational course***, it is **not unreasonable** that you find yourself needing more time to digest the materials and build the skills. The harder you work in this course, the easier you may find in subsequent years and job interviews.

10 Mapping Raw Marks to Letter Grades

- For each grading unit, you will receive a **raw mark score** (not necessarily out of 100).
- The **weighted sum** of all grading units will be mapped to its letter grade.
  - Check the common **Grades and Grading Schemes**.
  - e.g., Say there are only two grading units: Exam (60%) and Lab1 (40%).
    Receiving 150 marks (out of 200) for Exam and 2 marks (out of 3) for Lab1 leads to a letter grade B (based on the weighted sum $\frac{150}{200} \times 60 + \frac{2}{3} \times 40 \approx 71.7$).

11 Available Help

- Office hours held by your **section instructor**
- For programming assignments, you can also find the contact information of TAs on the Common eClass Site.
  - TA hours are on-demand: contact a TA when you find the need.
  - TAs will try to arrange a Zoom session with you at a mutually-agreed time.