EECS2011: Fundamentals of Data Structures

Section X – Winter 2023

Last Updated: February 23, 2023

Feb. 23: Test and Assignment Dates Changed after Reading Week

Subject to Changes until: January 23, 2023

Course Syllabus

1 Course Policies

To ensure a smooth, fair, and effective in-person delivery of this course:

1. **Team Work Encouraged for Labs/Assignments**: You will be able to gain **full** marks for all **labs** as long as submission attempts are made by the corresponding submission deadlines.

   Your submitted labs, though awarded **full** marks automatically, will still be graded and given detailed feedback (i.e., compilation and testing results). **The grading results and feedback are intended to exemplify how the actual programming tests will be graded.** Therefore, it would be your best interest in submitting work representing 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 labs (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 Jackie 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 labs**. Be advised that **later scheduled (written and programming) tests will be based on these labs**, 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 risk **poor performance** in subsequent tests and the exam.

2. **No Team Work Allowed for Scheduled Tests**: All **written & programming tests** are to be completed **individually** (i.e., **team work is forbidden**).

3. **Plagiarism**: When submitting each of your **written tests** and **programming tests**, you claim that it is **solely** your work. It is considered as **an 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**, so please be fully responsible for your learning.

4. **Meeting Lab/Test Deadlines**: **Stringent deadlines** are imposed on all scheduled **written tests** (to be completed and submitted via eClass), as well as scheduled **programming tests** and **labs** (to be submitted via the *web submit* to the EECS server). An in-person **exam** will be scheduled by the registrar office to take place during the **exam period**. It is your responsibility for meeting all deadlines.

5. **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, lab assignments access and deadlines); and **2)** attending lectures, submitting lab assignments, and taking scheduled tests in time.

   No lab deadline extensions or deferred tests will be accommodated.

2 Instructor

- Chen-Wei (Jackie) Wang
  - Contact: jackie@eecs.yorku.ca (https://www.eecs.yorku.ca/~jackie/)
  - Office Hours: 15:30 – 16:30, Tue, Wed, Thu; or by appointments.
    Campus Office: Lassonde Building, Room 2043 [19, D5 in the Keele campus]
    Virtual Office: https://yorku.zoom.us/my/jackie.loves.oxford
  - You are welcome to visit the office hours via Zoom, but please understand that priorities will be given to those showing up in the campus office.

3 Venues

- In-Class Lectures
  - 10:00 – 11:30, Mondays & Wednesdays
    VC 135 (Vanier College) [C6/56 on the Keele Campus Map]

  Note. In rare circumstances (e.g., extreme weather), in-person classes may be canceled: either a live Zoom lecture will take place, or lecture videos will be released.

4 eClass Site

- A single site for Section X: https://eclass.yorku.ca/course/view.php?id=64852

5 Study Materials

- There will be no textbooks for this course. Study your instructor's lecture materials:
  - The lectures page:
    https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS2011_W23
- For a thorough review on OOP in Java, consider the study materials for:
  - EECS2030-F21 (remote delivery):
    https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS2030_F21
  - EECS2030-F22 (in-person delivery):
    https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS2030_F22
- Here are some optional reference books:
  - Data Structures and Algorithms in Java, 6th Edition (2014), Wiley
    Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser
  - Algorithms, 4th Edition (2011), Addison-Wesley Professional
    Robert Sedgewick, Kevin Wayne [https://algs4.cs.princeton.edu/home/]

6 Available Help Resources

- Jackie's office hours [regular; request appointments if needed]
- TA office hours [on demand via Zoom; see eClass for TA's contact info]

7 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

8 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.

9 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.

10 Grading Scheme

		Subtotal
4 Assignments (2.5% each)	10%	30%
2 Programming Tests (10% each)	20%	
2 Written Tests (10% each)	20%	70%
Exam (Cumulative)	50%	

11 Final Exam: Cumulative & Substantial

- Your final exam will be *cumulative*: it will cover **all** study materials.
  - It will be an opportunity for you to synthesize topics that are connected.
- Therefore, your final exam will be the *most substantial* grading component.
  - It's meant to assess how competently you can apply the learned concepts and skills.
  - The best preparation for it is to constantly review and reflect on topics.

12 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.

13 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 $\frac{150}{200} \times 60 + \frac{2}{3} \times 40 \approx 71.7$).

14 Attendance of Classes: Encouraged & Rewarding

- There are 23 upcoming in-class lectures in total (2 classes $\times$ 12 weeks $-$ first class).
- Attending classes (in-time & focused) is an **indispensable** part of your learning.
- Despite it being your responsibility, Jackie would encourage you to attend classes by the following rewarding scheme:
  - Attendance will be taken **randomly** (via iClicker) on X classes ($10 \leq X \leq 23$)
    $\Rightarrow$ Attendance will be checked somewhere between every class and every other class.
  - Each attendance will be checked briefly (e.g., for a few minutes) at sometime between **5 minutes** after class starts and **5 minutes** before class ends.
  - **No** makeup attendance will be considered if you miss an attendance check (e.g., because you arrive late or leave early).
    $\Rightarrow$ Please **always** have the iClicker launched on your computer or mobile device.
  - At the end of the semester, say you attended Y classes:

if	$Y < \lfloor 50\% \cdot X \rfloor$	$\rightarrow$	no bonus
elseif	$Y \geq \lfloor 50\% \cdot X \rfloor$	$\rightarrow$	.5% bonus
elseif	$Y \geq \lfloor 60\% \cdot X \rfloor$	$\rightarrow$	1% bonus
elseif	$Y \geq \lfloor 70\% \cdot X \rfloor$	$\rightarrow$	2% bonus
elseif	$Y \geq \lfloor 80\% \cdot X \rfloor$	$\rightarrow$	4% bonus
elseif	$Y \geq \lfloor 90\% \cdot X \rfloor$	$\rightarrow$	5% bonus

  - For examples:

$X = 23$ (check at every class)

if	$Y < 11$	$\rightarrow$	no bonus
elseif	$Y \geq 11$	$\rightarrow$	.5% bonus
elseif	$Y \geq 13$	$\rightarrow$	1% bonus
elseif	$Y \geq 16$	$\rightarrow$	2% bonus
elseif	$Y \geq 18$	$\rightarrow$	4% bonus
elseif	$Y \geq 20$	$\rightarrow$	5% bonus

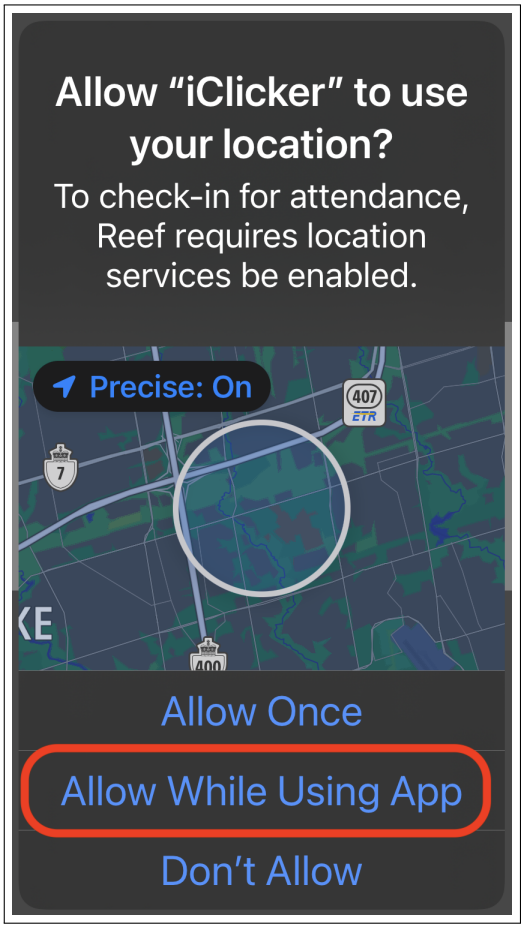
$X = 10$ (check at every other class)

if	$Y < 5$	$\rightarrow$	no bonus
elseif	$Y \geq 5$	$\rightarrow$	.5% bonus
elseif	$Y \geq 6$	$\rightarrow$	1% bonus
elseif	$Y \geq 7$	$\rightarrow$	2% bonus
elseif	$Y \geq 8$	$\rightarrow$	4% bonus
elseif	$Y \geq 9$	$\rightarrow$	5% bonus

- The above rewarding scheme **only** applies to in-class lectures.
- The allowable quota for you to miss classes, so as to get a particular bonus, already accommodates valid excuses (e.g., sick, family emergency).

  Therefore, **no** excuses will be considered for missing classes.

- The instructor reserves the right to **cancel** your bonus if you attend classes but cause distractions (e.g., talking, using devices for irrelevant activities) to the instructor and/or to other students.
- What should I do to set up the iClicker for attendance checks?
  - Refer to this starter guide (to install iClicker on your mobile device):
    `https://lthelp.yorku.ca/polling-students/iclicker-student-app-quick-start-guide`
    When creating an iClicker account, be sure to supply your **student number** and **...@my.yorku.ca** email (you are responsible for not receiving the bonus if an invalid student number or email is supplied).
  - Ignore the first section "**For Courses using eClass integration**".
  - Follow these sections:
    - "**For Courses not using eClass integration**"
    - "**Add Your Instructor iClicker Course**":
      Search for "`EECS2011-X (W23) - Fundamentals of Data Structures`".
    - "**Respond to Polls**"
  - When launching iClicker, it is critical that you allow iClicker to use your location; otherwise you will not be able to join the course and take attendance.



15 Semester Calendar

Figure 1 summarizes the schedule of required work items:

- Attend the scheduled in-class lectures on Monday and Wednesday (10:00 to 11:20).
- For weeks where a (written or programming) test is scheduled:
  - The test will occur during the **~~Wednesday~~ Monday class time in WSC**.
    [rooms to be confirmed]
  - Lecture videos will be released to compensate the missed class.
    Note. Specific details for each test will be announced in advance.
- All announced (written & programming) test dates are **fixed**, unless postponed due to unforeseen factors (e.g., weather condition).
- The assignment release dates may be *flexible*: they will be released as we get to the relevant topics in lectures. However, once released, you will be given an appropriate amount of time for completion.

Figure 1: EECS2011-X W23 Semester Calendar – Expected Work Items

16 Coverage of Tests

Tentatively, referencing the semester calendar in Figure 1 (p8):

- Written Test 1 covers Lectures 1 – ~~8~~ 7
- Written Test 2 covers Lectures ~~9~~ 8 – ~~16~~ 17
- Programming Test 1 covers Assignment 1 and Assignment 2
- Programming Test 2 covers Assignment 3 ~~and Assignment 4~~

17 Weekly Schedule

In the time table below, each cell denotes a 30-minutes interval.

- Cell 10:00 denotes the interval starting at 10:00 and ending at 10:30.
- For example, office hours (on Tuesdays, Wednesdays, and Thursdays) start at 15:30 and end at 16:30.

	Monday	Tuesday	Wednesday	Thursday	Friday
10:00	EECS2011-X Lecture VC 135		EECS2011-X Lecture (VC 135) **or** Test (WSC)		
10:30					
11:00					
11:30					
12:00					
12:30					
13:00					
13:30					
14:00					
14:30					
15:00					
15:30		Office Hours (In-Person or Zoom)			
16:00					
16:30					
17:00					
17:30					
18:00					
18:30					
19:00					
19:30					
20:00					

18 (Tentative) Lecture Topics

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

- Generics
- Abstract Classes and Interfaces
- Asymptotic Analysis of Algorithms
- (Iterative) Sorting: Selection, Insertion
- Linked Lists: Singly-Linked vs. Double-Linked
- Abstract Data Types (ADTs)
- Stacks, Queues, Deque
- Recursion
- Searching (Binary Search) and Sorting (MergeSort, QuickSort), Running Time, Correctness via Induction
- General Trees, Terminology, Tree Traversals, Binary Trees (BTs)
- Binary Search Trees (BSTs)
- Balanced BSTs, AVL Trees
- Heap, HeapSort
- ADT: Priority Queue