EECS2030: Advanced Object Oriented Programming

Sections B & E – Fall 2021

Last Updated: September 9, 2021

Course Syllabus

1 Course Policies

To ensure a smooth, fair, and effective online delivery of this course:

1. **Plagiarism**: When submitting each of your **labs** or **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 **will** examine **all** submitted code, and suspicious submissions 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.

2. **Online Submission/Assessment**: **Stringent deadlines** are imposed on all **scheduled written & programming tests** (to be completed and submitted via eClass), as well as **labs** (to be submitted via the *web submit* to the EECS server). An **exam** is scheduled online (via eClass) with **stringent timing requirements** (start time, duration, and end time to be announced by the registrar office).

   All announced deadlines are in the Eastern Time Zone (Toronto time). Students on a different time zone must figure out the corresponding local time.

   Students are responsible for **taking proactive steps and/or seeking assistance** well in advance to ensure that their technical setup (e.g., stable internet connection, a computer which does not freeze sporadically) allows them to complete and submit each assessment item (written test, programming test, lab, exam) in time.

   Rationales for this policy are to: **urge** students with technical issues to take steps or seek assistance to fix/improve them (otherwise, how can they benefit from the online setting in the first place?); and **discourage** students trying to take an unfair advantage (e.g., a student ignorant of the submission deadline or starting late may claim technical failure to have an extension, a student who has already seen the exam questions may claim network/computer failure in order to gain extra time or a deferred exam).

   When it comes to assessments, your instructor's priorities are fairness and academic integrity.

3. **No Team Work**: All **labs** and **written & programming tests** are to be developed and completed **individually** (i.e., **team work is forbidden**). This is meant for avoiding students having difficulties finding a suitable teammate and disputes between teammates (e.g., non-responsiveness, overdue progress, last-minute notice of withdrawal): the online nature of this course would only exacerbate these problems.

4. **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)** studying lecture videos, attending Q&A sessions, taking quizzes, and submitting lab assignments in time.

   No lab deadline extensions or deferred tests will be accommodated.

2 Academic Integrity

Labs

- All labs are to be completed ***individually***: no group work is allowed.

  TAs will perform thorough checks on **all** lab submissions: convincingly suspicious submissions will be reported to the Lassonde Student Service for a ***formal investigation*** immediately.

- To protect yourself from ending up a submission that is suspiciously similar to someone else's, you want to ***avoid***:

  - Discussing code-level details about labs/project with anyone.
  - Discussing concrete steps about your solution or someone's solution.
  - Sharing any part(s) of your code (e.g., file transfer via email, discord channels, SMS, screen sharing via Zoom) at any stage of your development.
  - Giving or receiving instructions about what exactly you should type for a fragment of code.

    (e.g., it is *acceptable* to ask about how to write a loop in general, but *unacceptable* to ask about how to write a loop specifically for solving a problem related to the assignment).

- The best ways to help your fellow students are clarifying instructions and showing them how to use breakpoints/debugger.

Written Tests & Programming Tests

- All written and programming tests, as well as the final exam, are to be completed ***individually***: no group work is allowed.

  TAs will perform thorough checks on **all** programming test submissions: convincingly suspicious submissions will be reported to the Lassonde Student Service for a ***formal investigation*** immediately.

- It is considered ***a breach of academic honesty*** if:

  - You collaborate with someone on completing a written or programming test during any stage of your development.
  - After you have attempted the written or programming test and before that test is closed, share your test questions with someone.

Reporting Cases

Enforcing the policy of academic honesty not only maintains the ***standard*** of the course, but also ensures ***fairness*** among all students in the class. If you have sufficient reasons to believe that cases of violation are present, let the instructor know and confidentiality will be maintained.

3 Instructors

– Chen-Wei (Jackie) Wang [Section B & Section E]

- Contact: jackie@eecs.yorku.ca (http://www.eecs.yorku.ca/~jackie/)
- Virtual Office: https://yorku.zoom.us/my/jackie.loves.oxford
- Office Hours: 15:00 – 16:00 (EST), Tue, Wed, Thu; or by Appointments.

4 eClass Site

– There is a single eClass site shared by Sections B & E:

https://eclass.yorku.ca/eclass/course/view.php?id=55896

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#EECS2030_F21

– For extra practice with Java, consider this tutorial series (created for EECS1022-W21): https://www.eecs.yorku.ca/~jackie/teaching/tutorials/index.html#java_from_scratch_w21

– Here are some optional reference textbooks:

- Introduction to Programming in Java: An Interdisciplinary Approach (2nd Ed.)

6 Available Help Resources

– Course forum on the common (B&E) eClass site

– Your instructor's office hours

– Scheduled lab sessions (you can attend any, multiple, or all of them to ask TA questions)

– Weekly Q&A sessions (held by the instrutor)

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/EECS 1021 3.00 or LE/EECS 1020 3.00 or LE/EECS 1022 3.00 or LE/EECS 1720 3.00

8 Course Description

This course continues the separation of concern theme introduced in all of its three predecessors (the legacy course EECS 1020, or the new EECS 1021, EECS 1022). While EECS1021/1022 focuses on the client concern, this course focuses on the concern of the implementer. Hence, rather than using an API (Application Programming Interface) to build an application, the student is asked to implement a given API.

Topics include implementing classes (utilities/non-utilities, delegation within the class definition, documentation and API generation, implementing contracts), aggregations (implementing aggregates versus compositions and implementing collections), inheritance hierarchies (attribute visibility, overriding methods, abstract classes versus interfaces, inner classes); generics; building graphical user interfaces (GUI) with an emphasis on the Model-View-Controller (MVC) design pattern; recursion; searching and sorting (including quick and merge sorts); linked lists; and stacks and queues. The coverage also includes a few design patterns.

Three lecture hours and weekly (90-minutes) laboratory sessions. Lab tests and in-class tests are integral parts of the assessment process in this course.

Throughout the course an Integrated Development Environment (IDE), such as Eclipse, and a testing framework, such as JUnit, are used.

9 Course Learning Outcomes (CLOs)

Upon completion of the course, students are expected to develop their:

- CLO1 Implement an Application Programming Interface (API).
- CLO2 Test the implementation.
- CLO3 Document the implementation.
- CLO4 Implement aggregations and compositions.
- CLO5 Implement inheritance.
- CLO6 Use recursion.
- CLO7 Implement linked lists.
- CLO8 (Informally) prove that recursive algorithms are correct and terminate.
- CLO9 (Informally) analyse the running time of (recursive) algorithms.

10 Grading Scheme

		Subtotal
Lab0 Part 1 & Part 2 (Review on OOP): 1.25% each	2.5%	15%
Lab1 – Lab5 (OOP in Java): 2.5% each	12.5%	
Programming Test 1	7%	27%
Programming Test 2	10%	
Programming Test 3	10%	
Written Tests 1 – 3: 6% each	18%	58%
Exam (Comprehensive)	40%	

11 Section B vs. Section E

- Labs, programming & written tests, and exam are common to both sections (B & E).

  Instructions will be posted on the common (B & E) eClass site.

  ∗ Labs will be submitted via the web submit.

  ∗ Programming & written tests and exam will be submitted via eClass.

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:
  - Lecture Videos [≈ 3 hours]

    Optional: Schedule Labs, Q&A sessions, Office Hours
- "Out-of-Class" Hours:
  - Completing Lab 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.

13 Mapping Raw Marks to Letter Grades

According to the Common Grading Scheme for Undergraduate Faculties approved by Senate:

Letter Grade	Grade Point	Interpretation
A+	9	Exceptional
A	8	Excellent
B+	7	Very Good
B	6	Good
C+	5	Competent
C	4	Fairly Competent
D+	3	Passing
D	2	Marginally Passing
E	1	Marginally Failing
F	0	Failing

- For each grading unit, you will receive a **raw mark score** (not necessarily out of 100).
- The <u>**weighted sum**</u> of all grading units will be mapped to its letter grade.

  e.g., Say there are only two grading units: Exam (60%) and Lab1 (40%).

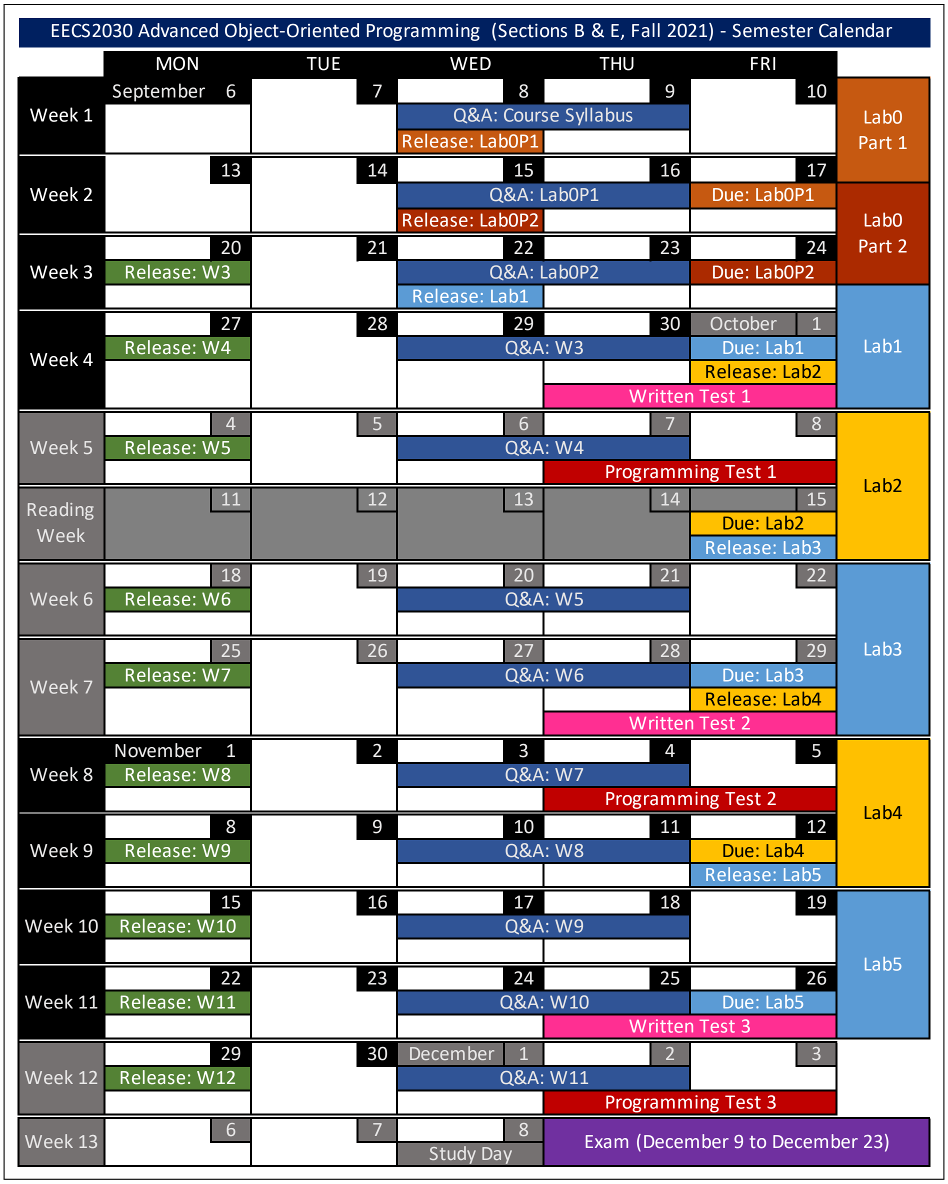
  A student receiving 150 marks (out of 200) for Exam and 2 marks (out of 3) for Lab1 has:

  Weighted sum: $\frac{150}{200} \times 60 + \frac{2}{3} \times 40 \approx 71.7$

  Letter grade: B

14 Semester Calendar

- Figure 1 summarizes the schedule of required work items:
  - Pre-recorded lectures are released on Mondays (except Week 1 and Week 2).
  - Optional Q&A sessions (for lectures) are held during the scheduled class times: Wednesdays for Section B and Thursdays for Section E.
  - Labs are released on Fridays (except Lab0 and Lab1) and due on Fridays.
  - A written test lasts for **20 minutes** during a 24-hour period between Thursday & Friday.
  - A programming test lasts for **90 minutes** during a 24-hour period between Thursday & Friday.
- For the precise times of the above work items, see Section 15 (submission time of labs and tests) and Section 16 (lecture Q&A time and scheduled lab/TA time).



EECS2030 Advanced Object-Oriented Programming (Sections B & E, Fall 2021) - Semester Calendar

	MON	TUE	WED	THU	FRI	
Week 1	September 6	7	8 Q&A: Course Syllabus Release: Lab0P1	9	10	Lab0 Part 1
Week 2	13	14	15 Q&A: Lab0P1 Release: Lab0P2	16	17 Due: Lab0P1	Lab0 Part 2
Week 3	20 Release: W3	21	22 Q&A: Lab0P2 Release: Lab1	23	24 Due: Lab0P2	
Week 4	27 Release: W4	28	29 Q&A: W3	30 Written Test 1	October 1 Due: Lab1 Release: Lab2	Lab1
Week 5	4 Release: W5	5	6 Q&A: W4	7 Programming Test 1	8	Lab2
Reading Week	11	12	13	14	15 Due: Lab2 Release: Lab3	
Week 6	18 Release: W6	19	20 Q&A: W5	21	22	Lab3
Week 7	25 Release: W7	26	27 Q&A: W6	28 Written Test 2	29 Due: Lab3 Release: Lab4	
Week 8	November 1 Release: W8	2	3 Q&A: W7	4 Programming Test 2	5	Lab4
Week 9	8 Release: W9	9	10 Q&A: W8	11	12 Due: Lab4 Release: Lab5	
Week 10	15 Release: W10	16	17 Q&A: W9	18	19	Lab5
Week 11	22 Release: W11	23	24 Q&A: W10	25 Written Test 3	26 Due: Lab5	
Week 12	29 Release: W12	30	December 1 Q&A: W11	2 Programming Test 3	3	
Week 13	6	7	8 Study Day	Exam (December 9 to December 23)		

Figure 1: EECS2030-B&E F21 Semester Calendar – Expected Work Items

15 Lab & Test Submission Time

- **Lab Submission Time**
  - Each lab will be ***released***, by **17:00 EST**, on its corresponding week day.
  - Each lab will be ***due***, at **14:00 EST**, on its corresponding **Friday**.
  - **No** late lab submissions will be accepted.
  - Submissions must be through the web submit.
  - Lab assignments are only graded offline, but **not** during the scheduled lab sessions.
- **Programming Test Submission Time**
  - For both Sections B & E, each programming test will be ***open*** for submission at **14:00 EST** on its corresponding **Thursday** and ***closed*** for submission at **14:00 EST** on its corresponding **Friday**.
  - During this 24-hour period, a single attempt lasting for **90 minutes** is allowed.
  - Submissions must be through eClass.
- **Written Test Submission Time**
  - For both Sections B & E, each written test will be ***open*** for submission at **14:00 EST** on its corresponding **Thursday** and ***closed*** for submission at **14:00 EST** on its corresponding **Friday**.
  - During this 24-hour period, a single attempt lasting for **20 minutes** is allowed.
  - Submissions must be through eClass.

For the precise dates of the above work items, see Section 14.

16 Scheduled Q&A and Lab Time

– In the time table below, each cell denotes a 30-minutes interval. For examples:

- Cell 11:30 denotes the interval starting at 11:30 and ending at 12:00.
- The fact that the Q&A session for Section B (on Wednesday) occupies 3 cells indicates that it lasts for 1.5 hours (starting at 11:30 and ending at 13:00).
- The fact that the Q&A session for Section E (on Thursday) occupies 3 cells indicates that it lasts for 1.5 hours (starting at 13:00 and ending at 14:30).

	Monday	Tuesday	Wednesday	Thursday	Friday
9:30					
10:00					
10:30					
11:00					
11:30		EECS2030-B&E TA Q&A (optional)	EECS2030-B Lecture Q&A or Test		
12:00					
12:30					
13:00		EECS2030-B&E TA Q&A (optional)		EECS2030-E Lecture Q&A or Test	
13:30					
14:00					
14:30	EECS2030-B&E TA Q&A (optional)				
15:00		office hours (Zoom)			
15:30					
16:00					
16:30					
17:00					
17:30				EECS2030-B&E TA Q&A (optional)	
18:00					
18:30					EECS2030-B&E TA Q&A (optional)
19:00					
19:30					
20:00					
20:30					
21:00					

– For Section B and Section E:

- Both scheduled lecture time slots
  - ∗ **11:30 – 13:00 on Wednesdays**
  - ∗ **13:00 – 14:30 on Thursdays**

  are used to hold (optional, Zoom) Q&A sessions to answer your questions related to the lecture materials.

  You are welcome to attend **any** of them to ask questions related to lectures/tutorials.
- All scheduled lab sessions are *optional*: you are welcome to attend **any** of them to ask questions related to lab assignments.

Remark. For both kinds of Q&A sessions, it is completely your decision on attending one, more, or none of them. However, I would **not** advise that you skip all of them, unless you are absolutely confident with the course materials.

17 (Tentative) Weekly Lecture Topics

Lecture videos are being actively recorded, so the order of topics below are subject to changes.

Week	**Topics**
1, 2	• Review of OOP in Java: classes, objects, methods • Tracing Object Creations and Method Calls: Eclipse Debugger vs. Paper • Inferring Classes and Methods from JUnit Test Cases • Declaring and Manipulating Reference-Typed, Multi-Valued Attribtes
3	• Exceptions • Testing for Exceptions • Test Driven Development (TDD)
4	• Object Equality • Call-by-Value • Aggregation and Composition
5	• Aggregation and Composition • Inheritance (motivating example, alternative designs, code reuse)
Reading Week	
6	• Inheritance (expectations, polymorphism, dynamic binding)
7	• Inheritance (type casts, polymorphic arguments and return values)
8	• Abstract Class and Interfaces
9	• Generics • Recursion
10	• Recursion
11	• Recursion • Asymptotic Analysis
12	• (Singly-)Linked Lists • Wrap-Up