EECS 1022 3.00: Programming for Mobile Computing

Sections M, N, O – Winter 2021

Course Syllabus

1 Course Policies

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

1. **Plagiarism**: When you submit your **labs** or **programming test**, 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, diagrams) during **any** stages 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 and programming) tests** (to be completed online on eClass), as well as **labs** (to be submitted to eClass *electronically* 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 and 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 deadline extension of labs 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 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 M & Section N]
  - Contact: jackie@eecs.yorku.ca (http://www.eecs.yorku.ca/~jackie/)
  - Virtual Office: https://yorku.zoom.us/my/jackie.loves.oxford
  - Office Hours: 1:00pm – 2:00pm, Mon, Tue, Wed; or by Appointments.
- Mufleh Al-Shatnawi [Section O]
  - Contact: mufleh@eecs.yorku.ca
  - Office Hours: By Appointments

4 eClass Sites

- There are three eClass sites:
  1. eClass site shared by Sections M,N,O: https://eclass.yorku.ca/eclass/course/view.php?id=6214
  2. eClass site for Sections M,N only: https://eclass.yorku.ca/eclass/course/view.php?id=6202
  3. eClass site for For Section O only: https://eclass.yorku.ca/eclass/course/view.php?id=6226
- If you are registered in Section M or Section N, check sites 1) and 2) regularly.
- If you are registered in Section O, check sites 1) and 3) regularly.

5 Study Materials

- There will be no textbooks for this course. Study your instructor's lecture materials:
  - For Sections M & N, see: **1)** the Section M,N eClass site; and **2)** the lectures page: https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS1022_W21
  - For Section O, see the Section O eClass site.
- For extra practice with Java, consider this tutorial series (created for EECS1021-W19): https://www.eecs.yorku.ca/~jackie/teaching/tutorials/index.html#java_from_scratch
- Here are some optional reference textbooks:
  - Introduction to Programming in Java: An Interdisciplinary Approach (2nd Ed.)
  - Introduction to Computer Science with Android (2nd Ed.) [Book Site]

6 Available Help Resources

- Course forum on the common (M,N,O) 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 your section instrutor)

7 Prerequisites

– LE/EECS1012 3.00 or LE/EECS1015 3.00

8 Course Description

This course provides a first exposure to object-oriented programming and enhances student understanding of key computing skills such as reasoning about algorithms, designing user interfaces, and working with software tools. It uses problem-based approach to expose the underlying concepts and an experiential laboratory to implement them. A mature mobile software infrastructure (such as Java and the Android programming environment) is used to expose and provide context to the underlying ideas.

Laboratory exercises expose students to a range of real-world problems with a view of motivating computational thinking and grounding the material covered in lectures. The lectures (two hours weekly) are supplemented by a three-hour weekly lab.

9 Course Learning Outcomes (CLOs)

Upon completion of the course, students are expected to:

CLO1 Understand software development within an object-oriented framework using a modern programming language and tool set.

CLO2 Use a set of computing skills such as reasoning about algorithms, tracing programs, test-driven development, and diagnosing faults.

CLO3 Explain and apply fundamental constructs in even-driven programs, including variables and expressions, control structures (conditionals/loops), and API usage.

CLO4 Write simple programs using a given software infrastructure, API, and tool chain.

CLO5 Gain exposure to a comprehensive mobile computing framework.

CLO6 Gain exposure to user interface design.

10 Satisfying CLOs: Eclipse vs. Android Studio

– Android Studio is a heavy-weighted environment assuming good knowledge about Java.

– In the *first 8 weeks*, we focus on **CLO1** – **CLO4**: OO programming, testing, and debugging in Eclipse, which **remains useful** in later courses (e.g., EECS2030, EECS2011).

– In the *final 4 weeks*, we focus on **CLO4** – **CLO6**, by applying the Java skills in developing Apps in Android Studio.

11 Grading Scheme

		Subtotal
Programming Test 1	5%	79%
Programming Test 2	8%	
Programming Test 3	10%	
Programming Test 4	10%	
3 Written Tests (4% each)	12%	
Exam (Comprehensive)	34%	
Lab0 (Java & Eclipse Intro)	1%	21%
Lab1 – Lab6 (Java OO Programming; 2.5% each)	15%	
Lab7 – Lab8 (Android App; 2.5%)	5%	

12 Sections M,N vs. Section O

- All labs, programming tests, and exam are common to all sections (M, N, O).
  - Instructions and submissions will be posted on the common (M,N,O) eClass site.
- Date and time of each written Test is common to all sections.
- Coverage of each written test is section-specific:
  - Sections M & N (Jackie) written tests take place on the Section M,N eClass site.
  - Section O (Mufleh) written tests take place on the Section O eClass site.

13 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 [2 hours]
  - Lab Tutorial Videos [1.5 – 2 hours]

    Optional: Schedule Labs, Q&A sessions, Office Hours
- "Out-of-Class" Hours:
  - Completing Lab Assignment, Studying for Lectures/Tests [6 to 9 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.

14 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).

– Weighted sum 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%).

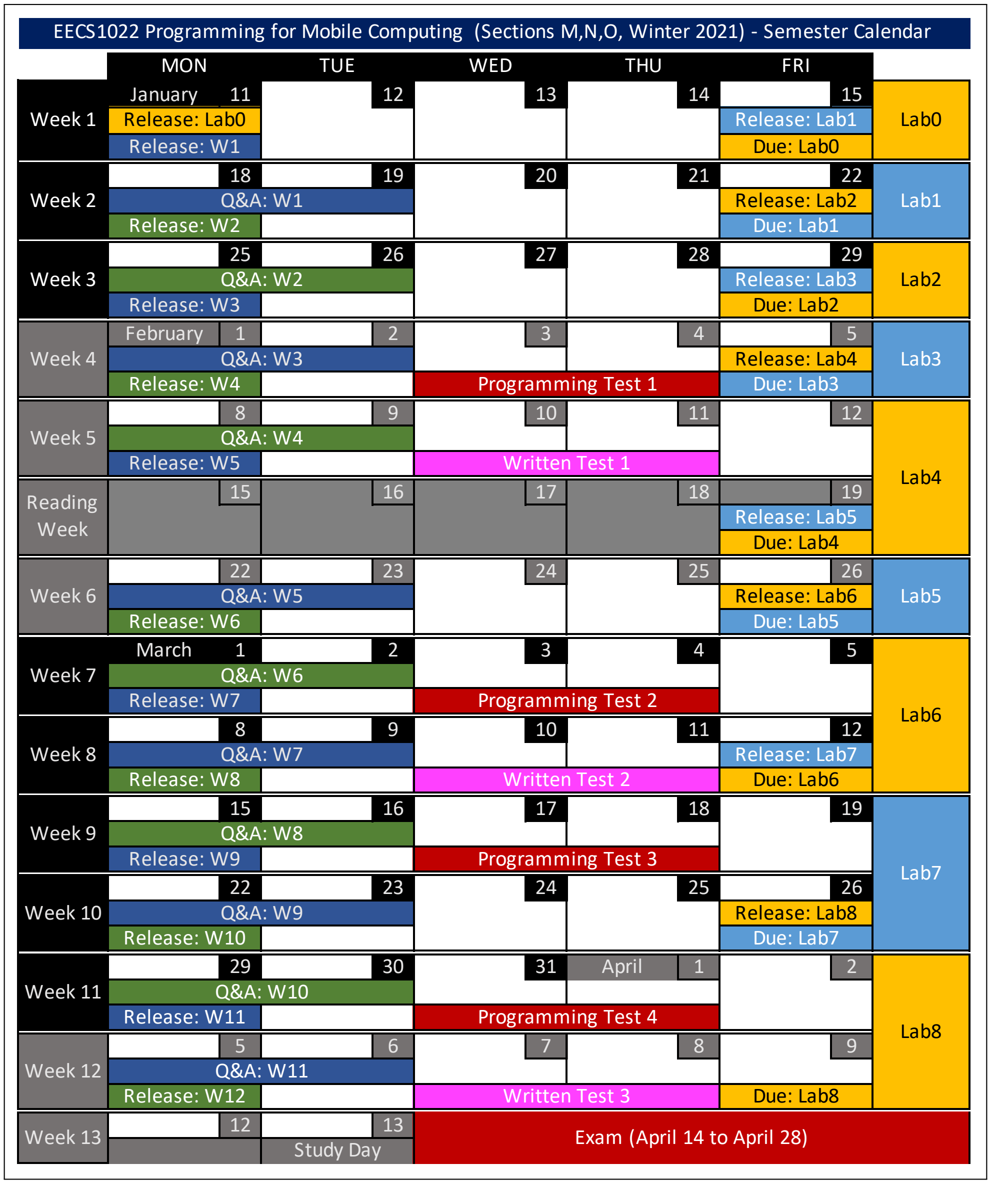
A student receiving 150 marks (out of 200) for Exam and 2 marks (out of 3) for Lab1 has:

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

Letter grade: B

15 Semester Calendar

- Figure 1 summarizes the schedule of required work items:
  - New pre-recorded lectures are released on Mondays.
  - Optional Q&A sessions (for lectures) are held during the scheduled class times (i.e., Mondays and Tuesdays for Section M,N; Mondays for Section O).
  - Labs are released/due on (most) Fridays.
  - Tests (written or programming) take place between (some) Wednesdays and Thursdays.
- For the precise time of the above work items, see Section 16 (submission time of labs and tests) and Section 17 (lecture Q&A time and scheduled lab/TA time).



EECS1022 Programming for Mobile Computing (Sections M,N,O, Winter 2021) - Semester Calendar

	MON	TUE	WED	THU	FRI	
Week 1	January 11 Release: Lab0 Release: W1	12	13	14	15 Release: Lab1 Due: Lab0	Lab0
Week 2	18 Q&A: W1 Release: W2	19	20	21	22 Release: Lab2 Due: Lab1	Lab1
Week 3	25 Q&A: W2 Release: W3	26	27	28	29 Release: Lab3 Due: Lab2	Lab2
Week 4	February 1 Q&A: W3 Release: W4	2	3 Programming Test 1	4	5 Release: Lab4 Due: Lab3	Lab3
Week 5	8 Q&A: W4 Release: W5	9	10 Written Test 1	11	12	Lab4
Reading Week	15	16	17	18	19 Release: Lab5 Due: Lab4	
Week 6	22 Q&A: W5 Release: W6	23	24	25	26 Release: Lab6 Due: Lab5	Lab5
Week 7	March 1 Q&A: W6 Release: W7	2	3 Programming Test 2	4	5	Lab6
Week 8	8 Q&A: W7 Release: W8	9	10 Written Test 2	11	12 Release: Lab7 Due: Lab6	
Week 9	15 Q&A: W8 Release: W9	16	17 Programming Test 3	18	19	Lab7
Week 10	22 Q&A: W9 Release: W10	23	24	25	26 Release: Lab8 Due: Lab7	
Week 11	29 Q&A: W10 Release: W11	30	31 Programming Test 4	April 1	2	Lab8
Week 12	5 Q&A: W11 Release: W12	6	7 Written Test 3	8	9 Due: Lab8	
Week 13	12	13 Study Day	Exam (April 14 to April 28)			

Figure 1: EECS1022 W21 Semester Calendar – Expected Work Items

16 Lab & Test Submission Time

– Lab Submission Time

- Each lab[1] will be ***released***, by **11:59pm EST**, on its corresponding **Friday**.
- Each lab will be ***due***, at **11:59pm EST**, on its corresponding **Friday**.

  Policy on **late lab submission**:

Overdue Hours (h)	Penalty (Percentages off)
$0 < h \leq 24$ (i.e., by 11:59pm EST on Saturday)	30%
$24 < h \leq 48$ (i.e., by 11:59pm EST on Sunday)	60%
$h > 48$	100%

- Submissions must be through the ***common (M,N,O) eClass site***

  (by attaching the required files).
- Lab assignments are only graded offline, but **not** during the scheduled lab sessions.

– Programming Test Submission Time

- Each programming test will be ***open*** for submission, at **02:00pm EST**, on its corresponding **Wednesday**.
- Each programming test will be ***closed*** for submission, at **02:00pm EST**, on its corresponding **Tursdays**.
- Submissions must be through the ***common (M,N,O) eClass sites*** .
- During the 24-hour submission period, there is a **single attempt of 90 minutes** for you to complete the test.

– Written Test Submission Time

- Each written test will be ***open*** for submission, at **02:00pm EST**, on its corresponding **Wednesday**.
- Each written test will be ***closed*** for submission, at **02:00pm EST**, on its corresponding **Tursdays**.
- Submissions must be through the ***section-specific (M,N or O) eClass sites*** .
- During the 24-hour submission period, there is a **single attempt of 30 minutes** for you to complete the test.

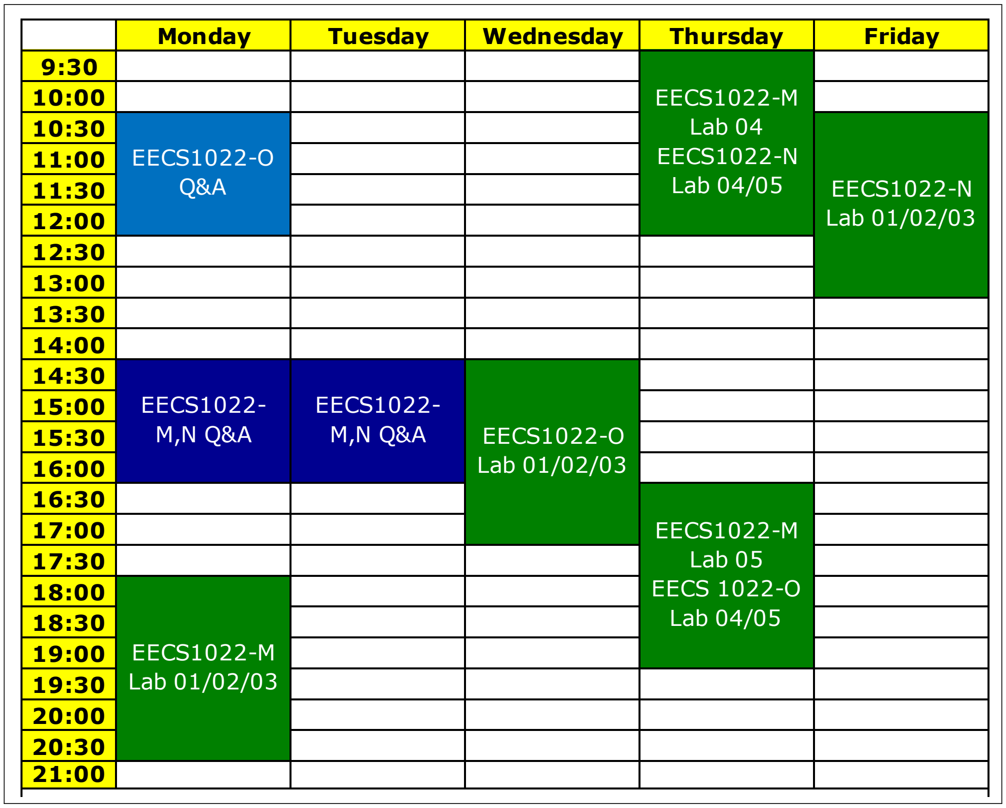
For the precise dates of the above work items, see Section 15.

[1] The only exception is Lab0, released on the beginning of the term, which is a Monday.

17 Scheduled Q&A and Lab Time

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

- Cell 14:30 denotes the interval starting at 14:30 and ending at 15:00.
- The fact that the Q&A session of Sections M,N (on Monday) occupies 4 cells indicates that it lasts for 2 hours (starting at 14:30 and ending at 16:30).



	Monday	Tuesday	Wednesday	Thursday	Friday
9:30				EECS1022-M Lab 04 EECS1022-N Lab 04/05	
10:00					
10:30	EECS1022-O Q&A				EECS1022-N Lab 01/02/03
11:00					
11:30					
12:00					
12:30					
13:00					
13:30					
14:00					
14:30	EECS1022- M,N Q&A	EECS1022- M,N Q&A	EECS1022-O Lab 01/02/03		
15:00					
15:30					
16:00					
16:30				EECS1022-M Lab 05 EECS 1022-O Lab 04/05	
17:00					
17:30					
18:00	EECS1022-M Lab 01/02/03				
18:30					
19:00					
19:30					
20:00					
20:30					
21:00					

– For Section M and Section N:

- Both schedule lecture time slots **14:30 – 16:30 on Mondays & Tuesdays** 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.

– For Section O:

- The schedule lecture time slot **10:30 – 12:30 on Mondays** is used to hold (optional, Zoom) Q&A sessions to answer your questions related to the lecture materials.

– All scheduled lab sessions are *optional*: you are welcome to attend any of them to ask questions related to lab assignments.

18 (Tentative) Weekly Lecture Topics

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

Week #	Topics
1	• Administrative • Elementary Programming
2	• Elementary Programming (continued) • Conditionals
3	• Conditionals (continued) • Reading API and Selecting (Static) Methods • Java API: `Math`, `String` (calling methods on strings)
4	• Loops: `for` vs. `while` • Loops: String Manipulations via Conditional & Loops (strings are like arrays of characters) • Loops and Arrays
5	Loops and Arrays (methods with array inputs and/or outputs)
Reading Week	
6	OO: classes vs. objects, primitive attributes, methods, primitive type vs. reference type OO: static vs. non-static attributes OO: static vs. non-static methods
7	• OO: Single-Referenced Attributes/Variables, Navigation via the Dot Notation • OO: Implementing & Using Methods with Conditionals • OO: Reference Aliasing
8	• OO: Multi-Referenced Attributes/Variables via Arrays, Navigation via the Dot Notation • OO: Implementing & Using Methods with Conditionals & Loops • OO: Reference Aliasing
9	Two-Dimensional Arrays and Nested Loops
10	Two-Dimensional Arrays and Nested Loops
11	• Reading API and Selecting (Non-Static) Methods • Java API: `ArrayList`, `HashMap` • Model-View-Controller Pattern in Android App Development
12	• OO: Multi-Referenced Attributes/Variables via `ArrayList`, `HashMap` • OO: Navigation via the Dot Notation, Methods with Conditionals & Loops