

EECS2030: ADVANCED OBJECT ORIENTED PROGRAMMING

Sections E & F – Fall 2024

GRADING SCHEME SUBJECT TO CHANGES UNTIL: WEDNESDAY, SEPTEMBER 18, 2024

LAST UPDATED: OCTOBER 15, 2024

- Changes on the release/due dates for: Lab2 – Lab5

COURSE SYLLABUS

1	<u>COURSE POLICIES</u>	3
2	<u>PREREQUISITES</u>	4
3	<u>INSTRUCTOR</u>	4
4	<u>VENUES</u>	4
5	Sections E & F Coordinated	5
6	<u>ECLASS SITE</u>	5
7	<u>STUDY MATERIALS</u>	5
8	<u>AVAILABLE HELP RESOURCES</u>	5
9	<u>COURSE DESCRIPTION</u>	6
10	<u>COURSE LEARNING OUTCOMES (CLOs)</u>	6
11	<u>GRADING SCHEME</u>	7
12	<u>FINAL EXAM: CUMULATIVE & SUBSTANTIAL</u>	7
13	<u>MAPPING RAW MARKS TO LETTER GRADES</u>	8
14	<u>EXPECTED WEEKLY WORKLOAD</u>	8
15	<u>MISSED TESTS</u>	8
16	<u>ACCOMMODATION</u>	8
17	<u>ATTENDANCE OF CLASSES: ENCOURAGED & REWARDING</u>	9

18 <u>SEMESTER CALENDAR</u>	11
19 <u>COVERAGE OF TESTS</u>	12
20 <u>WEEKLY SCHEDULE</u>	12
21 <u>(TENTATIVE) LECTURE TOPICS</u>	13

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 receive **full** marks as long as submission attempts are made by the corresponding submission deadlines.

Your submitted labs/assignments, 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**).

When a scheduled test takes place between different lab sessions, 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.

3. **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.**

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 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

3 INSTRUCTOR

- Chen-Wei (JACKIE) Wang
 - Contact: jackie@eecs.yorku.ca (<https://www.eecs.yorku.ca/~jackie/>)
 - Office Hours:
 - * In-Person: 15:00 – 16:00, Tuesdays & Thursdays
 - * Zoom: 15:00 – 16:00, Mondays
 - * by appointments (Zoom or In-Person)
- Campus Office: Lassonde Building, Room 2043 [19, D5 in the Keele campus]
Virtual Office: <https://yorku.zoom.us/my/jackie.loves.oxford>

4 VENUES

- Section E
 - In-Class Lectures
 - * 13:00 – 14:30, Tuesdays & Thursdays
LSB 103 (Life Science Building) [C4/90 on the Keele Campus]
 - Scheduled Labs
 - * 8:30 – 10:00, Wednesdays
WSC 105/106/108 (William Small Centre) [D4/15 on the Keele Campus]
- Section F
 - In-Class Lectures
 - * 10:00 – 11:30, Tuesdays & Thursdays
LSB 103 (Life Science Building) [C4/90 on the Keele Campus]
 - Scheduled Labs
 - * 16:00 – 17:30, Wednesdays
WSC 105/106/108 (William Small Centre) [D4/15 on the Keele Campus]

5 Sections E & F Coordinated

- Both sections will:
 - share the same grading scheme and (lab & lecture) materials;
 - be given assessments (tests & the exam) at the same level of difficulty.
- However:
 - You **must** attend scheduled tests during the lab session in which you are enrolled.
 - You **may** choose to attend the lecture of the other section, subject to the capacity constraint of the lecture room.
 - Lectures of the two sections may **not** be conducted at the exact same pace.

6 ECLASS SITE

- Common to both sections: <https://eclass.yorku.ca/course/view.php?id=130400>
- Specific to Section E: <https://eclass.yorku.ca/course/view.php?id=119804>
- Specific to Section F: <https://eclass.yorku.ca/course/view.php?id=119807>

7 STUDY MATERIALS

- There will be no textbooks required for this course. Study your instructor's lecture materials:
 - The lectures page:
https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS2030_F24
- 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.)

8 AVAILABLE HELP RESOURCES

- Jackie's office hours [regular; request appointments if needed]
- Scheduled lab sessions [attend any of them to ask TA and/or Jackie questions]
- TA office hours [on demand via Zoom; see eClass for TA's contact info]

9 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.

10 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.

11 GRADING SCHEME

		SUBTOTAL
Lab0 Part 1 & Part 2 (Review on OOP): 1.25% each	2.5%	10%
Lab1 – Lab5 (OOP in Java): 1.5% each	7.5%	
Programming Test 1	6%	28%
Programming Test 2	8%	
Programming Test 3	14%	
Written Test 1	4%	62%
Written Test 2	8%	
Exam (Cumulative)	50%	

12 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.

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 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: Schedule Labs, Office Hours
- “Out-of-Class” Hours:
 - Completing labs/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.

15 MISSED TESTS

If you missed a scheduled test with a valid reason, you are required to prepare the necessary documents and visit the instructor’s office **within a week** after the scheduled test time. We will discuss the alternatives, depending on your situation.

16 ACCOMMODATION

If you are registered with the office of student accessibility services, you are required to visit the instructor’s office hour **within the first two weeks** of the semester.

- We will discuss how to best accommodate your need, e.g., test time.
- The setup required by the in-lab tests is hard to be satisfied by the office location.
- Instead, we will accommodate your need by having you take tests in an EECS lab.
- If agreed, we will need you to inform the office of this and cancel your test bookings.

17 ATTENDANCE OF CLASSES: ENCOURAGED & REWARDING

- There are 23 upcoming in-class lectures in total ($2 \text{ classes} \times 12 \text{ weeks} - \text{first class}$).
- 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) $X = 10$ (check at every other 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
  
```

```

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
  
```

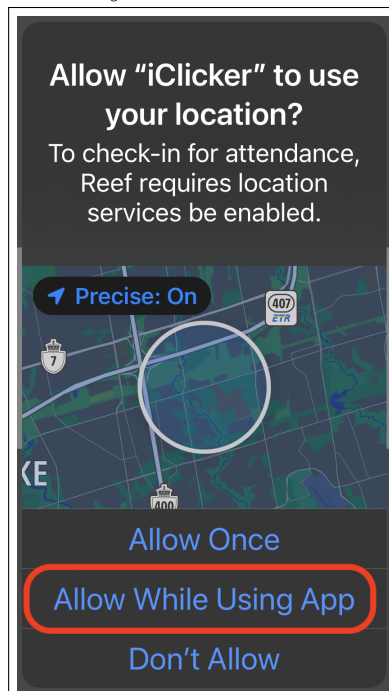
- The above rewarding scheme **only** applies to in-class lectures.
- The allowable quota for you to miss classes, so as to get a particular bonus, already accommodates valid excuses (e.g., sick, family emergency).

Therefore:

- once the attendance-taking window expired, **no** late responses will be accepted;
 - **no** reasons will be considered for missing attendance checks.
- The instructor reserves the right to **cancel** your bonus if, e.g.:
 - you just wait to be taken attendance and leave the class shortly after;
 - 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 “EECS2030-E&F (F’24) - Advanced OOP”.
 - * “**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.



18 SEMESTER CALENDAR

Figure 1 summarizes the schedule of required work items:

- Attend the scheduled in-class lectures.
 - All tests take place during the lab sessions you are enrolled in:
 - A written test lasts for ≈ 30 minutes.
 - A programming test lasts for ≈ 60 minutes.
- Note.** Specific details for each test will be announced in advance.
- If no test is scheduled, lab attendance is optional: TAs and/or Jackie will be there to answer your questions related to lab exercises and/or other course materials.
 - When a lab exercise is not allocated any scheduled lab session, please seek help via:
 - * Jackie's office hours
 - * TA's on-demand Zoom sessions
- [details sent in due course]

EECS2030 Advanced Object-Oriented Programming (Sections E & F, Fall 2024) - Semester Calendar														
	MON		TUE		WED		THU		FRI					
	September	2		3		4		5		6				
Week 1					Fall'24 Starts						Lab0 Part 1			
			Release: Lab0P1				Lecture 1							
Week 2		9		10		11		12		13	Lab0 Part 2			
			Lecture 2		Release: Lab0P2		Lecture 3			Due: Lab0P1				
Week 3		16		17		18		19		20	Lab1			
			Lecture 4				Lecture 5		Release: Lab1					
Week 4		23		24		25		26		27	Lab2			
			Lecture 6				Lecture 7							
Week 5		30	October	1		2		3		4	Lab2			
			Lecture 8		Written Test 1		Lecture 9		Release: Lab2					
Week 6		7		8		9		10		11	Lab3			
			Lecture 10		Prog Test 1		Lecture 11							
Reading Week		14		15		16		17		18	Lab4			
Week 7		21		22		23		24		25	Lab5			
			Lecture 12				Lecture 13		Release: Lab3					
Week 8		28		29		30		31	November	1	Lab4			
			Lecture 14		Prog Test 2		Lecture 15		Release: Lab4					
Week 9		4		5		6		7		8	Lab5			
			Lecture 16				Lecture 17		Drop Deadline					
Week 10		11		12		13		14		15				
			Lecture 18		Written Test 2		Lecture 19		Release: Lab5					
Week 11		18		19		20		21		22	Lab5			
			Lecture 20		Prog Test 3		Lecture 21							
Week 12		25		26		27		28		29	Lab5			
			Lecture 22				Lecture 23		Due: Lab5					
Week 13	December	2		3		4	Exam (December 5 to December 20)							
			Lecture 24		Study Day									

Figure 1: EECS2030-E&F F24 Semester Calendar – Expected Work Items

19 COVERAGE OF TESTS

Tentatively, referencing the semester calendar in Figure 1 (p11):

- Written Test 1 covers Lectures 1 – 7
- Written Test 2 covers Lectures 8 – 17
- Programming Test 1 covers Lab0 (Part 1 & Part 2) and Lab1
- Programming Test 2 covers Lab2 and Lab3
- Programming Test 3 covers Lab4

20 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 Mondays, Tuesdays, and Thursdays) start at 15:00 and end at 16:00.

	Monday	Tuesday	Wednesday	Thursday	Friday
8:30					
9:00			EECS2030 E Lab		
9:30					
10:00		EECS2030 F Lecture		EECS2030 F Lecture	
10:30					
11:00					
11:30					
12:00					
12:30					
13:00		EECS2030 E Lecture		EECS2030 E Lecture	
13:30					
14:00					
14:30					
15:00	Office Hour (Zoom)	Office Hour (In-Person)		Office Hour (In-Person)	
15:30					
16:00			EECS2030 F Lab		
16:30					
17:00					
17:30					

21 (TENTATIVE) LECTURE TOPICS

The order of topics is subject to changes.

Lectures	Topics
1 – 4	<ul style="list-style-type: none"> • 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 Attributes
5 – 6	<ul style="list-style-type: none"> • Exceptions
7 – 8	<ul style="list-style-type: none"> • Testing for Exceptions • Test Driven Development (TDD)
9 – 10	<ul style="list-style-type: none"> • Object Equality
11 – 12	<ul style="list-style-type: none"> • Call-by-Value • Aggregation and Composition
13 – 14	<ul style="list-style-type: none"> • Inheritance (motivating example, alternative designs, code reuse)
15 – 16	<ul style="list-style-type: none"> • Inheritance (expectations, polymorphism, dynamic binding)
17 – 18	<ul style="list-style-type: none"> • Inheritance (type casts, polymorphic arguments and return values)
19 – 20	<ul style="list-style-type: none"> • Abstract Class and Interfaces • Generics
21 – 22	<ul style="list-style-type: none"> • Recursion
23 – 24	<ul style="list-style-type: none"> • Recursion • Wrap-Up