

# EECS2030: ADVANCED OBJECT ORIENTED PROGRAMMING

Section F – Fall 2022

LAST UPDATED: SEPTEMBER 7, 2022

SUBJECT TO CHANGES UNTIL: SEPTEMBER 20, 2022

---

## COURSE SYLLABUS

1	<u>COURSE POLICIES</u>	2
2	<u>INSTRUCTOR</u>	3
3	<u>VENUES</u>	3
4	<u>ECLASS SITE</u>	3
5	<u>STUDY MATERIALS</u>	3
6	<u>AVAILABLE HELP RESOURCES</u>	3
7	<u>PREREQUISITES</u>	4
8	<u>COURSE DESCRIPTION</u>	4
9	<u>COURSE LEARNING OUTCOMES (CLOs)</u>	4
10	<u>GRADING SCHEME</u>	5
11	<u>MAPPING RAW MARKS TO LETTER GRADES</u>	5
12	<u>EXPECTED WEEKLY WORKLOAD</u>	6
13	<u>ATTENDANCE OF CLASSES: ENCOURAGED &amp; REWARDING</u>	6
14	<u>SEMESTER CALENDAR</u>	8
15	<u>COVERAGE OF TESTS</u>	9
16	<u>WEEKLY SCHEDULE</u>	9
17	<u>(TENTATIVE) LECTURE TOPICS</u>	10

# 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](mailto:jackie@eecs.yorku.ca) (<https://www.eecs.yorku.ca/~jackie/>)
  - In-Person Office Hours: 12:30 – 13:30 (EST), Tue & Thu; or by Appointments.  
Campus Office: Lassonde Building, Room 2043 [ 19, D5 in the Keele campus ]
  - Zoom Office Hours: 16:00 – 17:00 (EST), Mon & Wed; or by Appointments.  
Virtual Office: <https://yorku.zoom.us/my/jackie.loves.oxford>

## 3 VENUES

- In-Class Lectures
  - 8:30 – 10:00, Mondays  
LSB 106 (Life Science Building) [ C4/90 on the Keele Campus Map ]
  - 8:30 – 10:00, Wednesdays  
LAS B (Lassonde Building) [ D5/19 on the Keele Campus Map ]
- Scheduled Labs
  - 16:00 – 17:30, Tuesdays  
WSC 105/106/108 (William Small Centre) [ D4/15 on the Keele Campus Map ]

## 4 ECLASS SITE

- There is an eClass site for Section F:  
<https://eclass.yorku.ca/course/view.php?id=64498>

## 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\\_F22](https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS2030_F22)
- 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](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

- Jackie’s office hours
- Scheduled lab session (to ask TA and/or Jackie questions)

## 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% each	2%	12%
Lab1 – Lab5 (OOP in Java): 2% each	10%	
Programming Test 1	5%	25%
Programming Test 2	8%	
Programming Test 3	12%	
Written Tests 1 – 3: 6% each	18%	63%
Exam (Cumulative)	45%	

## 11 MAPPING RAW MARKS TO LETTER GRADES

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

Letter Grade	Range of Marks	Interpretation
A+	$\geq 90$	Exceptional
A	$\geq 80$	Excellent
B+	$\geq 75$	Very Good
B	$\geq 70$	Good
C+	$\geq 65$	Competent
C	$\geq 60$	Fairly Competent
D+	$\geq 55$	Passing
D	$\geq 50$	Marginally Passing
E	$\geq 45$	Marginally Failing
F	$\geq 40$	Failing

- 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.

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

## 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 [  $\approx 3$  hours ]
  - **Optional:** Schedule Labs, 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 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 ( $12 \leq X \leq 23$ )  
 $\Rightarrow$  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 **10 minutes** after class starts and **10 minutes** before class ends.  
 $\Rightarrow$  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)     $X = 12$  (check at every other class)

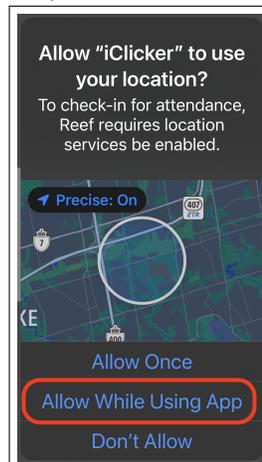
if	$Y < 11$	→	no bonus
elseif	$Y \geq 11$	→	.5% bonus
elseif	$Y \geq 13$	→	1% bonus
elseif	$Y \geq 16$	→	2% bonus
elseif	$Y \geq 18$	→	4% bonus
elseif	$Y \geq 20$	→	5% bonus

if	$Y < 6$	→	no bonus
elseif	$Y \geq 6$	→	.5% bonus
elseif	$Y \geq 7$	→	1% bonus
elseif	$Y \geq 8$	→	2% bonus
elseif	$Y \geq 9$	→	4% bonus
elseif	$Y \geq 10$	→	5% bonus

- The above rewarding scheme **only** applies to in-class lectures, **not** lab sessions.
- The allowable quota for you to miss classes, so as to get a particular bonus, already accommodates for 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?

- Please refer to this starter guide (to install iClicker on your mobile device):  
<https://lthelp.yorku.ca/polling-students/iclicker-student-app-quick-start-guide>
- 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 “LE/EECS2030 (Section F, F’22) - 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.



## 14 SEMESTER CALENDAR

Figure 1 summarizes the schedule of required work items:

- Attend the scheduled in-class lectures on Monday and Wednesday (8:30 to 9:50).
- All lab sessions take place on Tuesdays (16:00 to 17:20):
  - A written or programming, if scheduled, starts at 16:15.
    - \* A written test lasts for  $\approx$  30 minutes.
    - \* A programming test lasts for  $\approx$  65 minutes.

**Note.** Specific details for each test will be announced in advance.

- Otherwise, if no test is scheduled, the lab attendance is optional: TAs and Jackie will be there to answer your questions related to lab exercises and/or other course materials.

EECS2030 Advanced Object-Oriented Programming (Section F, Fall 2022) - Semester Calendar						
	MON	TUE	WED	THU	FRI	
Week 1	September 5	6	7 Release: Lab0P1 Lecture 1	8	9	Lab0 Part 1
Week 2	12 Lecture 2	13 Release: Lab0P2	14 Lecture 3	15	16 Release: Lab1 Due: Lab0P1	
Week 3	19 Lecture 4	20	21 Lecture 5	22	23 Due: Lab0P2	Lab0 Part 2
Week 4	26 Lecture 6	27 Written Test 1	28 Lecture 7 Due: Lab1	29	30 Release: Lab2	Lab1
Week 5	October 3 Lecture 8	4	5 Lecture 9 Prog Test 1	6	7	Lab2
Reading Week	10	11	12	13	14	
Week 6	17 Lecture 10	18	19 Lecture 11	20	21 Release: Lab3 Due: Lab2	
Week 7	24 Lecture 12	25	26 Lecture 13	27	28 Release: Lab4 Due: Lab3	Lab3
Week 8	31 Lecture 14	November 1 Written Test 2	2 Lecture 15	3	4	Lab4
Week 9	7 Lecture 16	8 Prog Test 2	9 Lecture 17	10	11	
Week 10	14 Lecture 18	15	16 Lecture 19	17	18 Release: Lab5 Due: Lab4	
Week 11	21 Lecture 20	22	23 Lecture 21	24	25	Lab5
Week 12	28 Lecture 22	29 Written Test 3	30 Lecture 23	December 1	2 Due: Lab5	
Week 13	5 Lecture 24	6 Prog Test 3	7 Study Day	Exam (December 8 to December 23)		

Figure 1: EECS2030-F F22 Semester Calendar – Expected Work Items

## 15 COVERAGE OF TESTS

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

- Written Test 1 covers Lectures 1 – 5
- Written Test 2 covers Lectures 6 – 13
- Written Test 3 covers Lectures 14 – 21
- Programming Test 1 covers Lab0 (Part 1 & Part 2) and Lab1
- Programming Test 2 covers Lab2 and Lab3
- Programming Test 3 covers Lab4 and Lab5

## 16 WEEKLY SCHEDULE

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

- Cell 8:30 denotes the interval starting at 8:30 and ending at 9:00.
- For example, the in-person office hours (on Tuesdays and Thursdays) occupies 2 cells indicates that it lasts for 1 hour (starting at 12:30 and ending at 13:30).

	Monday	Tuesday	Wednesday	Thursday	Friday
8:30	EECS2030-F Lecture LSB 106		EECS2030-F Lecture LAS B		
9:00					
9:30					
10:00					
10:30					
11:00					
11:30					
12:00					
12:30		Office Hours (In-Person)		Office Hours (In-Person)	
13:00					
13:30					
14:00					
14:30					
15:00					
15:30					
16:00	Office Hours (Zoom)	EECS2030-F Lab 1, 2, 3	Office Hours (Zoom)		
16:30					
17:00					

## 17 (TENTATIVE) LECTURE TOPICS

The order of topics is subject to changes.

Lectures	Topics
1 – 4	<ul style="list-style-type: none"> <li>• Review of OOP in Java: classes, objects, methods</li> <li>• Tracing Object Creations and Method Calls: Eclipse Debugger vs. Paper</li> <li>• Inferring Classes and Methods from JUnit Test Cases</li> <li>• Declaring and Manipulating Reference-Typed, Multi-Valued Attributes</li> </ul>
5 – 6	<ul style="list-style-type: none"> <li>• Exceptions</li> </ul>
7 – 8	<ul style="list-style-type: none"> <li>• Testing for Exceptions</li> <li>• Test Driven Development (TDD)</li> </ul>
9 – 10	<ul style="list-style-type: none"> <li>• Object Equality</li> </ul>
Reading Week	
11 – 12	<ul style="list-style-type: none"> <li>• Call-by-Value</li> <li>• Aggregation and Composition</li> </ul>
13 – 14	<ul style="list-style-type: none"> <li>• Inheritance (motivating example, alternative designs, code reuse)</li> </ul>
15 – 16	<ul style="list-style-type: none"> <li>• Inheritance (expectations, polymorphism, dynamic binding)</li> </ul>
17 – 18	<ul style="list-style-type: none"> <li>• Inheritance (type casts, polymorphic arguments and return values)</li> </ul>
19 – 20	<ul style="list-style-type: none"> <li>• Abstract Class and Interfaces</li> <li>• Generics</li> </ul>
21 – 22	<ul style="list-style-type: none"> <li>• Recursion</li> </ul>
23 – 24	<ul style="list-style-type: none"> <li>• Recursion</li> <li>• Wrap-Up</li> </ul>