EECS2030: Advanced Object Oriented Programming

Section E – Summer 2025

Grading Scheme Subject to Changes until: Monday, May 19, 2025

Last Updated: May 6, 2025

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 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/`)

    > Jackie believes that **in-person** communication is the *most effective* for attending to your questions/concerns related course materials and grading. When you receive slow or no responses to your email inquiries, it is most likely an indication that Jackie is happy to help you during his **in-person** office hours and/or appointments.

  - Office Hours:
    - 12:30 – 13:30, Tuesdays & Thursdays (**in-person**)
    - 13:00 – 14:00, Wednesdays
    - 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
    - 14:30 – 16:00, Tuesdays & Thursdays
      LAS B (Lassonde Building) [D5/19 on the Keele Campus]
  - Scheduled Labs
    - Lab01: 13:00 – 14:20, Fridays
    - Lab02: 14:30 – 15:50, Fridays
    - Lab03: 13:30 – 14:50, Fridays
      LAS 1006 (Lassonde Building) [D5/19 on the Keele Campus]

5 eClass Site

https://eclass.yorku.ca/course/view.php?id=131933

6 Study Materials

– There will be **no** textbooks required for this course. Study your instructor's lecture materials:

https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS2030_S25

– 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.)

7 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]

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): 0.5% each	1%	5%
Lab1 – Lab5 (OOP in Java): 0.8% each	4%	
Programming Test 0	2%	30%
Programming Test 1	6%	
Programming Test 2	8%	
Programming Test 3	14%	
Written Test 1	6%	65%
Written Test 2	9%	
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 **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.

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

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:
  - 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.

14 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 two days** after the scheduled test time. We will discuss the alternatives, depending on your situation.

15 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.

16 Attendance of Classes: Encouraged & Rewarding

- There are 23 upcoming in-class lectures in total (2 classes × 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 ($10 \leq X \leq 23$)
    ⇒ Attendance will be checked somewhere between every class and every other class.
    In a class where attendance is taken, **one more more checks** may be conducted: your attendance of that class will **not** count if you miss any of the checks. For example, if you wait for the first check to occur and choose to leave right afterwards, your attendance will not count as you may miss the subsequent check(s). That is, your attendance to a class will count **only if** you complete **all** checks.
  - Each attendance check will be conducted 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 missed a check because you, e.g.,
    - arrived late
    - left early
    - did not pay attention or was absent when the attendance check took place
  - Please **always** have the iClicker launched on your computer or mobile device:
    - There will be a sign-up sheet to accommodate the **(extremely) rare** occurrences of failed check-ins.
      You will be accommodated to sign on a sheet for **a maximum of 2 classes**.
      ⇒ You are **solely** responsible for resolving any technical issues that caused you to fail checking in via the installed iClicker.
      e.g., see: https://mhe.my.site.com/iclicker/s/article/How-to-Troubleshoot-Your-Connection-to-the-iClicker-Student-App

- At the end of the semester, say you attended Y classes:

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

- For examples:

$X = 23$ (check at every class)

```
if      $Y \geq 20.7$ → 5% bonus
elseif  $Y \geq 18.4$ → 4% bonus
elseif  $Y \geq 16.1$ → 2% bonus
elseif  $Y \geq 13.8$ → 1% bonus
elseif  $Y \geq 11.5$ → .5% bonus
elseif  $Y < 11.5$ → no bonus
```

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

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

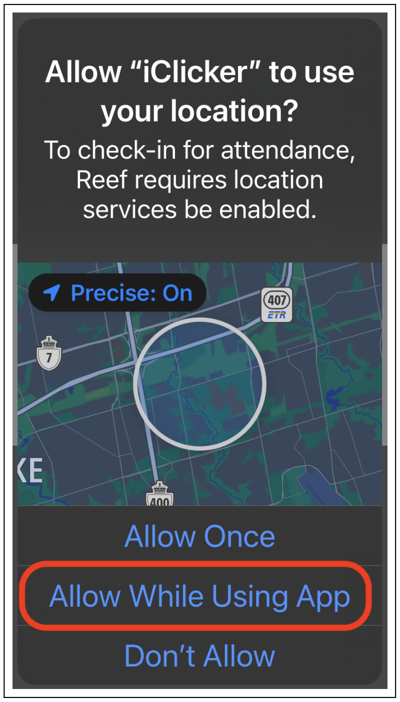
– The instructor reserves the right to **cancel** your bonus if, e.g.:

- 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**":
      `EECS2030-E (S'25) - 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.



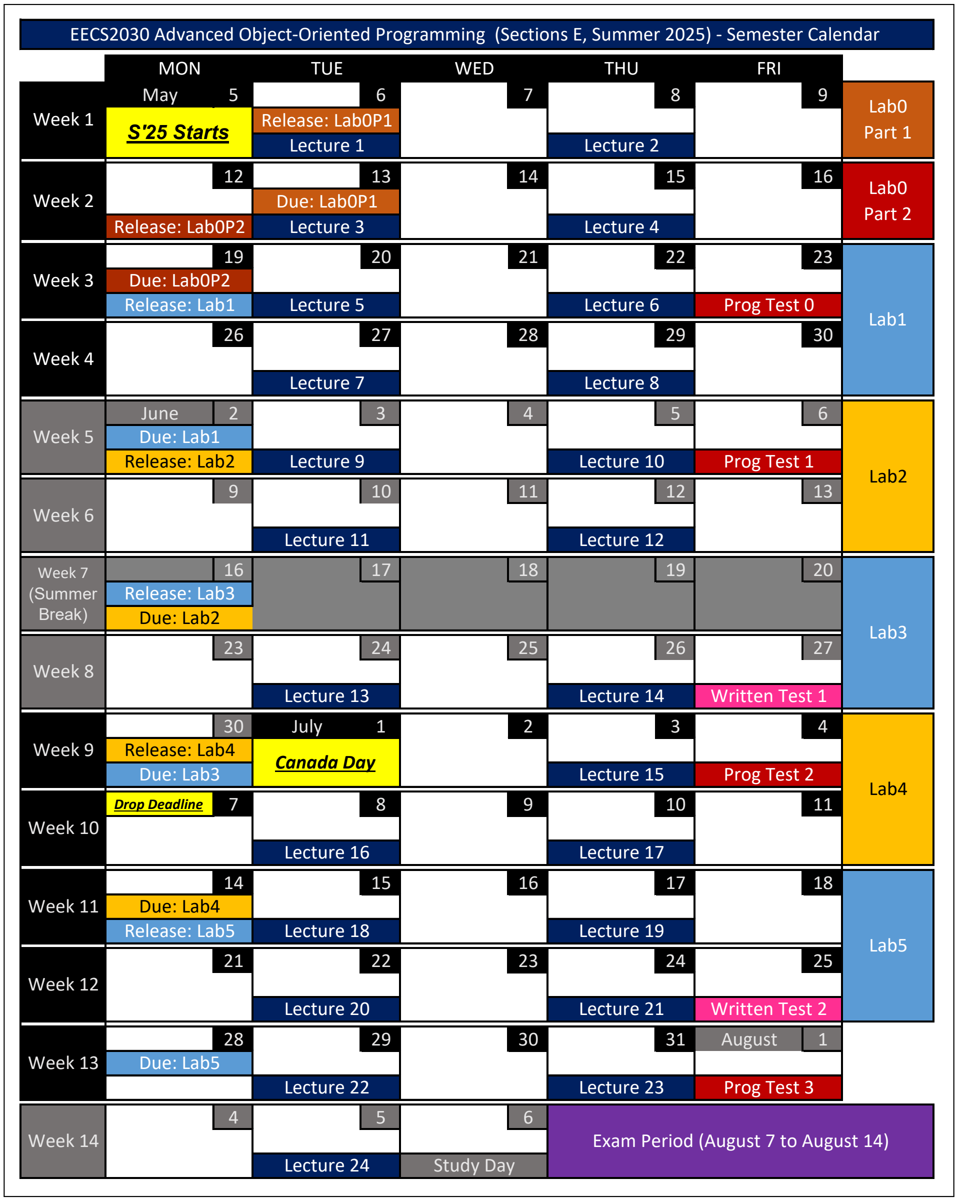
Allow "iClicker" to use your location?

To check-in for attendance, Reef requires location services be enabled.

17 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:
  - Specific details for each test (e.g., coverage, duration) 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, Summer 2025) - Semester Calendar

	MON	TUE	WED	THU	FRI	
Week 1	May 5 *S'25 Starts*	6 Release: Lab0P1 Lecture 1	7	8 Lecture 2	9	Lab0 Part 1
Week 2	12 Release: Lab0P2	13 Due: Lab0P1 Lecture 3	14	15 Lecture 4	16	Lab0 Part 2
Week 3	19 Due: Lab0P2 Release: Lab1	20 Lecture 5	21	22 Lecture 6	23 Prog Test 0	Lab1
Week 4	26	27 Lecture 7	28	29 Lecture 8	30	
Week 5	June 2 Due: Lab1 Release: Lab2	3 Lecture 9	4	5 Lecture 10	6 Prog Test 1	Lab2
Week 6	9	10 Lecture 11	11	12 Lecture 12	13	
Week 7 (Summer Break)	16 Release: Lab3 Due: Lab2	17	18	19	20	Lab3
Week 8	23	24 Lecture 13	25	26 Lecture 14	27 Written Test 1	
Week 9	30 Release: Lab4 Due: Lab3	July 1 *Canada Day*	2	3 Lecture 15	4 Prog Test 2	Lab4
Week 10	*Drop Deadline* 7	8 Lecture 16	9	10 Lecture 17	11	
Week 11	14 Due: Lab4 Release: Lab5	15 Lecture 18	16	17 Lecture 19	18	Lab5
Week 12	21	22 Lecture 20	23	24 Lecture 21	25 Written Test 2	
Week 13	28 Due: Lab5	29 Lecture 22	30	31 Lecture 23	August 1 Prog Test 3	
Week 14	4	5 Lecture 24	6 Study Day	Exam Period (August 7 to August 14)		

Figure 1: EECS2030-E S25 Semester Calendar – Expected Work Items

18 Coverage of Tests

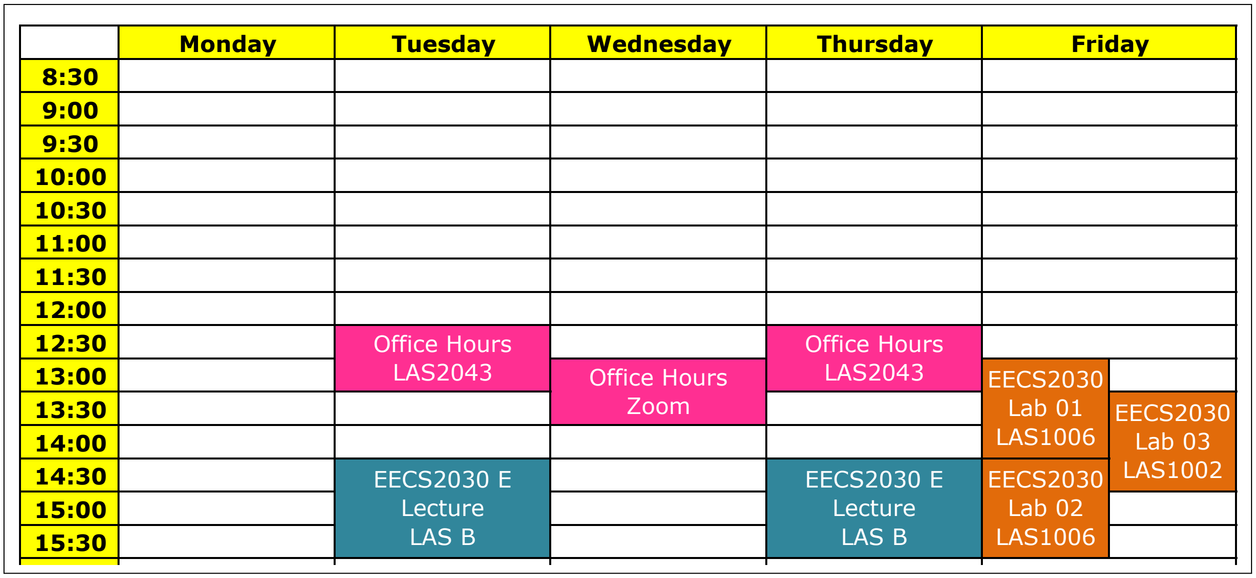
Tentatively, referencing the semester calendar in Figure 1 (p12):

- Written Test 1 covers Lectures 1 – 12
- Written Test 2 covers Lectures 13 – 19 (plus some earlier lectures)
- Programming Test 0 covers Lab0 (Part 1 & Part 2)
- Programming Test 1 covers Lab0 and Lab1
- Programming Test 2 covers Lab2 and Lab3
- Programming Test 3 covers Lab4 and Lab5

19 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, in-person office hours (on Tuesdays and Thursdays) start at 12:30 and end at 13:30.



	Monday	Tuesday	Wednesday	Thursday	Friday	
8:30						
9:00						
9:30						
10:00						
10:30						
11:00						
11:30						
12:00						
12:30		Office Hours LAS2043		Office Hours LAS2043		
13:00			Office Hours Zoom		EECS2030 Lab 01 LAS1006	
13:30						EECS2030 Lab 03 LAS1002
14:00						
14:30		EECS2030 E Lecture LAS B		EECS2030 E Lecture LAS B	EECS2030 Lab 02 LAS1006	
15:00						
15:30						

20 (Tentative) Lecture Topics

The order of topics is subject to changes.

Lectures	Topics
1 – 4	• 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
5 – 6	• Exceptions
7 – 8	• Testing for Exceptions • Test Driven Development (TDD)
9 – 10	• Object Equality
11 – 12	• Call-by-Value • Aggregation and Composition
13 – 14	• Inheritance (motivating example, alternative designs, code reuse)
15 – 16	• Inheritance (expectations, polymorphism, dynamic binding)
17 – 18	• Inheritance (type casts, polymorphic arguments and return values)
19 – 20	• Abstract Class and Interfaces • Generics
21 – 22	• Recursion
23 – 24	• Recursion • Wrap-Up