EECS4302: Compilers and Interpreters

Section A – Fall 2022

Last Updated: September 8, 2022

Subject to Changes until: September 20, 2022

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 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 (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
  14:30 – 16:00, Tuesdays & Thursdays
  R S129 (Ross Building South) [D5/28 on the Keele Campus Map]

4 eClass Site

- There is an eClass site for Section A:
  https://eclass.yorku.ca/course/view.php?id=64630

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#EECS4302_F22
- If you wish to review OOP in Java, consider the study materials for:
  - EECS2030 (Introductory OOP):
    https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS1022_W21
  - EECS2030 (Advanced OOP):
    https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS2030_F21
  - EECS2011 (Data Structures):
    https://www.eecs.yorku.ca/~jackie/teaching/lectures/index.html#EECS2011_W22
- Here are some optional reference textbooks:
  - Title: Engineering a Compiler
    Author: Keith D. Cooper & Linda Torczon
    Publisher: Morgan Kaufmann Publishers Inc.
    Edition: Second Edition (2012)
    ISBN: 978-0-12-088478-0
  - Classic Text (the "dragon" book):
    Title: Compilers: Principles, Techniques, and Tools
    Author: Alfred V. Aho, Monica S. Lam, Ravi Sethi & Jeffery D. Ullman
    Publisher: Addison-Wesley Longman Publishing Co., Inc.
    Edition: Second Edition (2006)
    ISBN: 978-0-321-48681-3

6 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/EECS2001 3.00; LE/EECS2011 3.00

7 Course Description

This course covers principles and design techniques for compilers and interpreters, purpose and design of domain-specific languages (DSLs), compiler organization, compiler writing tools, scanning, parsing, semantic analysis, code generation, and if time permitted, run-time storage organization, memory management, and optimization.

This course is a hands-on introduction to the design and construction of compilers and interpreters. The main purpose is to have students acquire the techniques of defining their own languages and implementing tools to suit their own needs. At the end of the course, students will understand the architecture of compilers and interpreters, their major components, how the components interact, and the algorithms and tools that can be used to construct the components.

Students will be assigned a semester-long project divided into phases: e.g., designing the grammar of a domain-specific language, using tools to generate its scanners and parsers, and applying suitable software design patterns (e.g., composite, visitor) to build abstract syntax trees (ASTs) of input programs and to perform semantic operations (e.g., evaluation, type checking, code generations). Generated programs may take different forms: e.g., C code, database schemas/procedures, theorems/functions to a proof assistant, specification language to a model checker.

8 Course Learning Outcomes (CLOs)

Upon completion of the course, students are expected to be able to:

CLO1 Compare and contrast General-Purpose Languages (GPLs) and Domain-Specific Languages (DSLs).

CLO2 Apply the theoretical understanding of, and use the relevant tools to generate, a lexical scanner.

CLO3 Apply the theoretical understanding of, and use the relevant tools to generate, a grammatical parser.

CLO4 Construct an abstract syntax tree & perform semantic operations on it.

CLO5 Communicate the design and implementation of a DSL and its associated tools.

9 Grading Scheme

3 Assignments (3% each)	9%
Project	18%
Programming Test	8%
5 In-Class Quizzes (4% each)	20%
Final Exam	45%

10 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+	≥ 90	Exceptional
A	≥ 80	Excellent
B+	≥ 75	Very Good
B	≥ 70	Good
C+	≥ 65	Competent
C	≥ 60	Fairly Competent
D+	≥ 55	Passing
D	≥ 50	Marginally Passing
E	≥ 45	Marginally Failing
F	≥ 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 $\frac{150}{200} \times 60 + \frac{2}{3} \times 40 \approx 71.7$).

11 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:
  - Weekly Lectures [≈ 3 hours]
    Optional: Office Hours
- "Out-of-Class" Hours:
  - Completing Assignments, Studying for Lectures/Tests [6 to 10.5 hours]
- You may find yourself needing more time to digest the materials and build the skills. The harder you work, the easier you may find in your future career endeavours.

12 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 ($12 \leq X \leq 23$)
    ⇒ 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.
    ⇒ Please always have the iClicker launched on your computer or mobile device.
  - At the end of the semester, say you attended Y classes:

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

- For examples:

  $X = 23$ (check at every 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

  $X = 12$ (check at every other class)

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

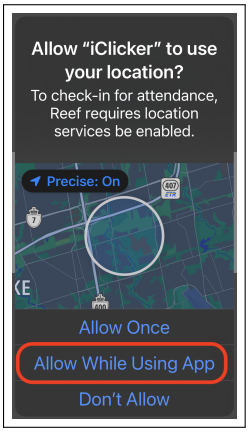
– The instructor reserves the right to **cancel** your bonus if you attend classes but cause distractions (e.g., talking, using devices for irrelevant activities) to the instructor and/or to other students.

– 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`

  When creating an iClicker account, please be sure to supply your **student number**.
- 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/EECS4302 (Section A, F'22) - Compilers and Interpreters`".
  * "**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.



13 Semester Calendar

Figure 1 summarizes the schedule of required work items:

- Attend the scheduled in-class lectures on Tuesdays and Thursdays (14:30 to 16:00).
- Each **in-class** written quiz, if scheduled, starts at 14:35 and lasts for ≈ 20 minutes.
  Note. Specific details for each quiz will be announced in advance.
- An **in-lab** programming test will be held on Saturday, October 29 (2 weeks before the drop deadline).
  - The exact time of the test will be surveyed to suit the schedule of most (if not all).
  - The test will last for ≈ 80 minutes.

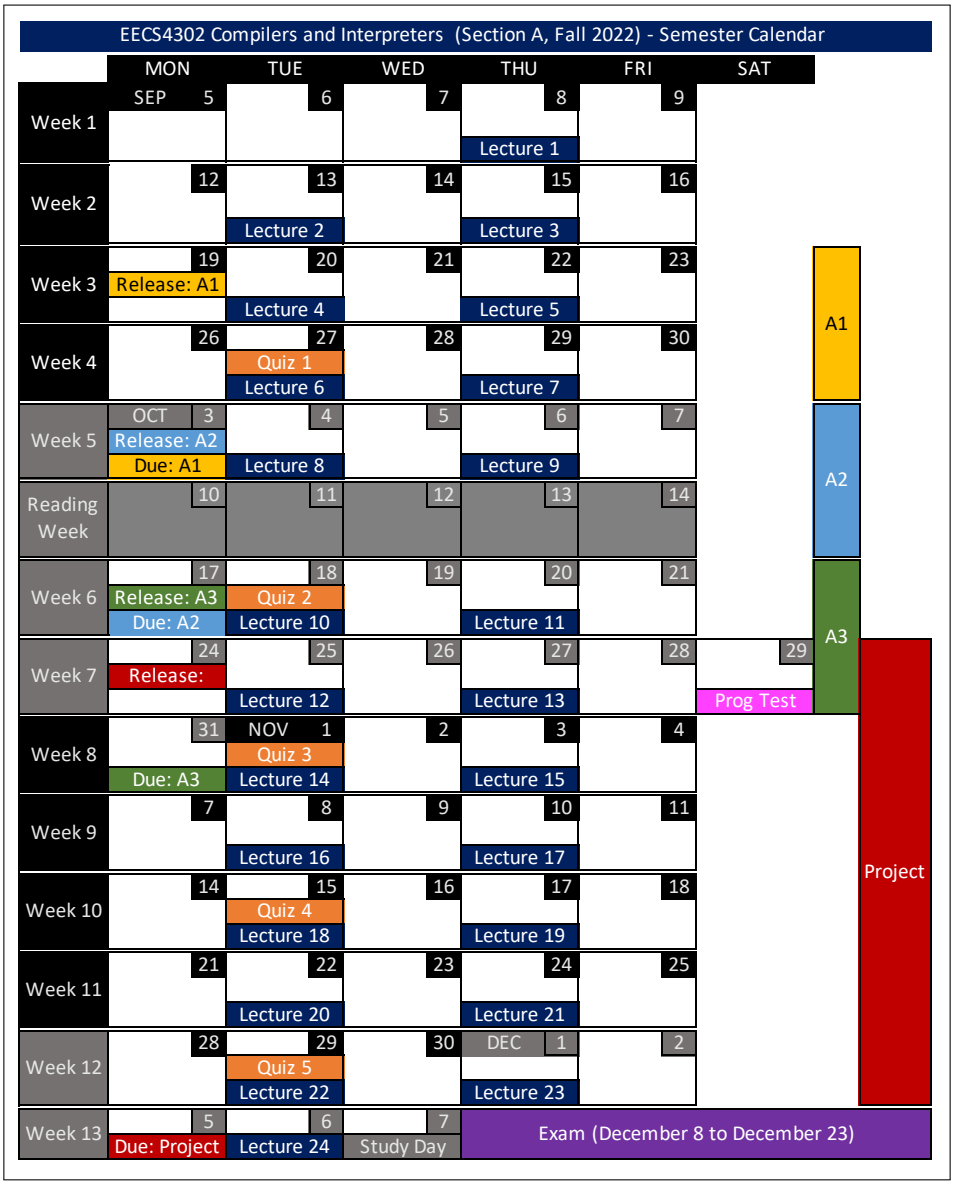


Figure 1: EECS4302-A F22 Semester Calendar – Expected Work Items

14 Coverage of Quizzes

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

- Quiz 1 covers Lectures 1 – 5
- Quiz 2 covers Lectures 6 – 9
- Quiz 3 covers Lectures 10 – 13
- Quiz 4 covers Lectures 14 – 17
- Quiz 5 covers Lectures 18 – 21
- Programming Test covers A1 – A3 and ANTLR4 tutorial

15 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					
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		EECS4302-A Lecture R S129		EECS4302-A Lecture R S129	
15:00					
15:30					
16:00	Office Hours (Zoom)		Office Hours (Zoom)		
16:30					

16 (Tentative) Topics to be Covered

- Overview of Compilation
- Lexical Analysis (Scanners)
- Syntax Analysis (Parsers)
- Use of a Parser Generator (ANTLR)
- OOP Design Patterns: Composite & Visitor
- Context-Sensitive Analysis
- Intermediate Representation
- Optimization