

EECS2030: ADVANCED OBJECT ORIENTED PROGRAMMING

Winter 2020

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

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; Recursions and Complexity analysis.

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.

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.

LAB TESTS

- In chosen lab sessions, you will be required to complete programming tasks (using an IDE) or writing tasks. These tests are designed to test your understanding of the taught concepts, as well as your mastery of using the programming tool to develop working solutions to given problems.
- These tests are based on lecture materials and lab exercises. For your preparation, instructions of the lab test will be distributed about one week in advance.
- For your submission to be assessed, you **must** submit *compilable* source code by the end of the test session. We do not have the TA resources to examine your written code, so you receive very low marks by submitting code that does not compile.

GRADING SCHEME

| | |
|-----------------|-----|
| Lab Test 0 | 5% |
| Lab Submissions | 10% |
| Lab Test #1 | 15% |
| Lab Test #2 | 15% |
| Lab Test #3 | 15% |
| Final Exam | 40% |

IMPORTANT INSTRUCTIONS

Submissions: Proper academic performance depends on students doing their work not only well, but on time. Accordingly, lab assignments for this course must be received on the due date specified for the assignment.

Late Penalty: All students are expected to complete their coursework (lab assignments etc) in a timely fashion. All late lab assignments should be given to the instructor. Late lab assignments etc. will be subject to a reduced grade of 20% of the maximum grade per

day late (where weekends will also count as two days). A mark of zero will be applied if the assignment is submitted after the solutions have been posted.

Missed Lab Tests and Exam: Students who miss the lab tests or examination (or leave the examination), due to documented illness, or due to prior arrangement with approval from the Course Instructor, an oral/written exam may be carried out in the end of the semester having all course contents covered so far. Otherwise, the mark will be zero. There will be no alternative dates scheduled to make up the assignments, quizzes, midterms, labs etc. If a student leaves the examination after 25 minutes, the exam will be graded and that mark will be applied. Absence to exam(s) must be accompanied by a formal and valid explanation within 24 hours of said absence. In addition to university policies on a deferral final exam, the granting of a deferral final exam also requires that the student has performed satisfactorily in the course according to the evaluation scheme in this Course outline, excluding the final examination for which deferral privileges are requested.

Labs: Students are expected to conduct their labs in a safe and professional manner. You will be briefed on the potential hazards and steps to mitigate any risk by a Technician or Teaching Assistant within the lab. If you have any doubt about anything related to your safety or another's safety, immediately ASK the technician or teaching assistant. Absence to labs must be accompanied by a formal and valid explanation within 24 hours of said absence (valid doctor's notes and by using said note, permission is given to the professor to inquire to its validity). It will be the student's responsibility to catch up on the material missed by speaking with classmates.

Copyright: Lectures and course materials, including power point presentations, outlines, and similar materials, are protected by copyright. You may take notes and make copies of course materials for your own educational use. You may not and may not allow others to reproduce or distribute lecture notes and course materials publicly for commercial and non-commercial purposes. If it can be shown that these terms were violated by you, your grade can be changed to an F even after the course is completed.

ACADEMIC INTEGRITY LINKS

- Senate Policy on Academic Honesty -
<http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/>
- Academic Integrity -
<http://lassonde.yorku.ca/academic-integrity>

STUDENT LINKS

- Student Rights and Responsibilities -
<http://oscr.students.uit.yorku.ca/student-conduct>
- Religious Observance -
<https://w2prod.sis.yorku.ca/Apps/WebObjects/cdm.woa/wa/regobs>
- Academic Accommodation for Students with Disabilities -
<http://secretariat-policies.info.yorku.ca/policies/academic-accommodation-for-students-with-disabilities-policy/>

- Student Accessibility Services (SAS) -
<https://accessibility.students.yorku.ca/>
- York University Racism Policy and Procedures -
<http://secretariat-policies.info.yorku.ca/policies/racism-policy-and-procedures/>
- York University's Policies on Sexual Violence -
<http://secretariat-policies.info.yorku.ca/policies/sexual-violence-policy-on/>
- York University's Policies on Gender/LGBTQ*/Positive Space -
<http://rights.info.yorku.ca/lgbtq/>

LAND ACKNOWLEDGMENT

We acknowledge our presence on the traditional territory of many Indigenous Nations. The area known as Tkaronto has been care taken by the Anishinabek Nation, the Haudenosaunee Confederacy, the Huron-Wendat, and the M̃tis. It is now home to many Indigenous Peoples. We acknowledge the current treaty holders, the Mississaugas of the New Credit First Nation. This territory is subject of the Dish With One Spoon Wampum Belt Covenant, an agreement to peaceably share and care for the Great Lakes region.

The Indigenous Framework for York University: A Guide to Action can be found here:
<http://indigenous.info.yorku.ca/>

Meaning of a land acknowledgement: <http://healthydebate.ca/opinions/indigenous-land-acknowledgements>

TENTATIVE COURSE CALENDAR

| W | Day | Date | Topics | Lab | Test |
|----------------------------------|-----|------------|--|-------------------|-------------------------|
| 1 | Mon | 6 Jan | - | Lab 0 Due 12 Jan | |
| | T,R | 7, 9 Jan | <ul style="list-style-type: none"> • Course Introduction • OO: classes, objects, methods | | |
| 2 | Mon | 13 Jan | - | Lab 1 Due 19 Jan | |
| | T,R | 14, 16 Jan | <ul style="list-style-type: none"> • OO: classes, objects, methods • OO: Aliasing • OO: Exceptions • DbC (precond., postcond., invariant) | | |
| 3 | Mon | 20 Jan | - | | Lab Test 0 ¹ |
| | T,R | 21,23 Jan | <ul style="list-style-type: none"> • DbC (precond., postcond., invariant) • OO: JUnit & Testing Strategies • Javadoc • OO: equals | - | |
| 4 | Mon | 27 Jan | - | Lab 2 Due 2nd Feb | |
| | T,R | 28,30 Jan | <ul style="list-style-type: none"> • OO: equals, comparable, hash code • static vs. non-static • Utility Classes (e.g., Math, Arrays) | | |
| 5 | Mon | 3 Feb | | | Lab Test 1 ² |
| | T,R | 4,6 Feb | <ul style="list-style-type: none"> • Use of Generics and Interface basics • Use of Collection • List: ArrayList • Set: HashSet • Map: HashMap, SortedMap, TreeMap | | |
| 6 | Mon | 10 Feb | - | Lab 3 Due 21 Feb | |
| | T,R | 11, 13 Feb | <ul style="list-style-type: none"> • Aggregation & Composition | | |
| Winter Reading Week: Feb 15 – 21 | | | | | |

¹Basics of Computational Thinking, Loops, Conditions, Arrays

²Classes, Objects, and Methods

| W | Day | Date | Topics | Lab | Test |
|----|------|------------|--|------------------|-------------------------|
| 7 | Mon | 24 Feb | - | Lab 4 Due 1 Mar | |
| | T,R | 25, 27 Feb | <ul style="list-style-type: none"> • Aggregation & Composition • Inheritance: Code Reuse | | |
| 8 | Mon | 2 Mar | - | | Lab Test 2 ¹ |
| | Tues | 3,5 Mar | Inheritance: Poly. & Dyn. Bin. | | |
| 9 | Mon | 9 Mar | - | Lab 5 Due Mar 21 | |
| | T,R | 10,12 Mar | <ul style="list-style-type: none"> • Inheritance: Poly. & Dyn. Bin. • Abstract Class & Interface (revisit) | | |
| 10 | Mon | 16 Mar | - | Lab 5 Due Mar 21 | |
| | T,R | 17, 19 Mar | Recursion: Programming | | |
| 11 | Mon | 23 Mar | - | | Lab Test 3 ² |
| | T,R | 24, 26 Mar | <ul style="list-style-type: none"> • Recursion: Programming • Recursion: Runtime & Correctness | | |
| 12 | Mon | 30 Mar | - | Lab #6 Due Apr 5 | |
| | T | 31 Mar | <ul style="list-style-type: none"> • Big O: Introduction | | |
| | R | 2 Apr | | | |

¹Aggregation & Composition

²Inheritance