# Concordia University
## Department of Computer Science and Software Engineering

## Advanced Programming Practices
## SOEN 6441 --- Winter 2024

### Contact Information
name:           Dr. Joey Paquet
office:          ER 10-19
phone:          (514) 848-2424 ext. 7953
office hours:   Mondays 11:00-13:00
e-mail:          joey.paquet@concordia.ca
www:            www.cse.concordia.ca/~paquet

### Schedule/Contacts

| lectures | LECT WW | ---J--- | 17:45-20:15 | H-531 | Paquet, Joey | joey.paquet@concordia.ca |
|---|---|---|---|---|---|---|
| laboratories | LAB WW WI | ---J--- | 20:30-22:20 | TBD | Sahota, Pankaj Deep | pankajdeepsahota2020@gmail.com |
| | LAB WW WJ | ---J--- | 15:45-17:35 | TBD | Jafarpour, Hamed | jafarpour.ha@gmail.com |
| | LAB WW WK | --W---- | 20:30-22:20 | TBD | Raza, Ahmad Faraz | ahmadfaraz123450@gmail.com |

### Calendar Description
Problems of writing and managing code. Managing complexity: programming process. Pragmatic Programming. Coding conventions, software documentation. Software configuration management. Advanced debugging techniques: program tracing, dynamic inspection and tools. Testing: coding techniques for testing software. Multithreading concurrency and distributed programming. Multi-language programming. A project. Laboratory: two hours per week.

### Rationale
Most students coming out of introductory programming courses know how to write simple programs written in a specific programming language. Other more advanced courses show them how to use the more advanced features of the language, thus reaching for a complete understanding of the syntactical constructs of a specific programming language. Industrial programming requires a lot more diversified skills than simple mastery of language syntax. Industrial programmers have to know how manage the complexity of their coding activities, install and use libraries, create reusable, documented, fault free and fault tolerant code. This course aims at broadening the knowledge of the students to these concepts, techniques and tools that are complementary to what is taught in standard programming courses; to improve the practical programming skills of students by emphasizing real-life aspects of programming that are not dealt with in regular introductory and advanced programming courses; as well as practical mastery of techniques and tools for the writing of superior quality code and complementary programming artifacts such as inline documentation, design patterns, and automated testing infrastructure.

### Prerequisite knowledge
Although this course does not officially have course prerequisites, it is taken for granted that the students are already proficient programmers that master the object-oriented programming paradigm. The lectures, exercises, examinations, and completion of the project will necessitate pre-existing programming skills. This course is not meant to teach you how to program, but rather how to extend your programming skills and apply them in a team project.

## Delivery
All the lectures and lab sessions will be pre-recorded and be made available online prior to the regular weekly lecture/lab time. The lectures and lab sessions will be delivered in-person. The instructor will provide a 2-hour weekly in-person/zoom office hours for individual consultation outside of the lecture time. The midterms and final examination will be held using a moodle quiz held in-person in a lab. All assignments/project demonstrations will be done on zoom using the shared screen feature. Students will have remote access to the lab infrastructure, though they may use their own computers and online resources to manage their project.

## Grading
**Project :** The project is to be tackled by teams of 5 members. It is divided into 3 practical and incremental assignments related to the project. Each assignment includes the delivery of an operational subset (i.e. and increment or build) of the final project. The project consists of a large program whose development involves most of the topics discussed in the lectures. Each build is graded independently of the other builds, following a grading scheme given prior to the due date. Each build will be presented orally in a practical demonstration done on zoom. All project assignments are marked as a team, and all team members get the same grades.

**Examinations :** Both the midterm and final examinations are held using online moodle quizzes whose goal is to individually test the comprehension of the material taught in class. The final examination covers the material covered in all the lectures. The numeric-to-letter grading conversion is made according the class average. You must have at least 50% total mark, as well as in the combined examinations mark in order to pass the course.

| Examinations (midterm: 20% + final: 30%) | 50% |
|---|---|
| Project build 1 | 15% |
| Project build 2 | 15% |
| Project build 3 | 20% |

## Laboratory
In addition to present the practical applications/examples pertaining to the material presented in the lectures, the laboratory instructor(s) will be there to help you on the project, most particularly on the use of the tools and libraries to be used for the implementation.

## Communication
All announcements to the students will be made through moodle messages. Thus it is very important for all students to regularly read e-mail coming into the address they have setup in moodle. All course material, as well as the weekly schedules and due dates, will be provided on the official moodle page for the course.

## Textbook
Walter Savitch. *Absolute Java.* Addison Wesley. Sixth Edition, 2015. (not mandatory)

## General Notes
In the event of circumstances beyond the instructor's or the University's control, the content and/or evaluation scheme of this course is subject to change. Students are expected to be aware of the University's Code of Conduct, what constitutes any violation to this code, and what are the consequences of violating this code.