Université Concordia University

GINA CODY
SCHOOL OF ENGINEERING
AND COMPUTER SCIENCE

SOEN 6431

Software comprehension and maintenance

Computer Science and Software Engineering

Fall 2023

Course Instructor:
Dr. Juergen Rilling
E-mail: juergen.rilling@concordia.ca

Office Hours:

In person:	***On Zoom:***
Days: Wednesdays	*Days: Sundays*
Time: 14.00 – 17.00	*Time: 20.00 – 21.00*
Location: ER 9-31	*Location: Zoom Link – please click*
(or by appointment)	

Tutorials: *N/A*
POD: *TBA*

Labs: *N/A*

Course Calendar Description:
The course addresses both technical and managerial views of software comprehension and software maintenance issues. Topics covered in this course include: cognitive models, software visualization, CASE tools, reverse engineering, static and dynamic source code analysis, software configuration management, and introduction to current research topics in software maintenance and program comprehension. A project is required.

Prerequisites: *N/A*
Co-requisites: *N/A*

Specific Knowledge and Skills Needed for this Course:
Students taking this course are expected to have sufficient knowledge of the following topics. Should you have difficulties in any of these topics, you are strongly encouraged to review them before the DNE deadline.

Intermediate or better programming skills (Java is a bonus),

Course materials
Required Textbook: *N/A*
Course material will be made available on Moodle

Grading Scheme

Grade distribution

Assignments (4)	*8%*
Case studies (3)	*12%*
Midterm	*35%*
Final Exam	*45%*

Grading scale

A curved grade distribution is applied. The approach uses these three steps: First, numeric scores are assigned to the students. The actual values are unimportant as long as the ordering of the scores corresponds to the ordering of how good the students are. In the second step these scores are converted to percentiles. Finally, the percentile values are transformed to grades according to a division of the percentile scale into intervals, where the interval width of each grade indicates the desired relative frequency for that grade.

Tentative Course Schedule

Week	***Content***
Week #1	*Course overview, Introduction Software Evolution*
Week #2	*Technical Debt , Software Traceability (Case study #1)*
Week #3	*Software analytics: source code analysis, program slicing (Assignment #1)*
Week #4	*Software Repositories (issue tracker, versioning, configuration), Mining software repositories*
Week #4	*Testing (Assignment #2)*
Week #5	*Impact Analysis and regression testing (Case study #2)*
	Fall Break
Week #6	*Midterm*
Week #7	*Program Comprehension*
Week #8	*Clone detection, Software Analytics (Assignment #3)*
Week #9	*Software Migration*
Week #10	*Special topics: Generative AI and software evolution (Case study #3)*
Week #11	*Special topics: Software Evolution and DevOps, DataOps, ModelOps (Assignment #4)*
Week #12	*Domain specific aspect of Software Evolution, Exam review*

Lab Details: N/A

Engineering Tools: Software Repositories

Details on assessment tools: Assignments and Quizzes will have to be submitted through MOODLE. **No late** submissions will be accepted.

Other information: Plagiarism:

The most common offense under the Academic Code of Conduct is plagiarism, which the Code defines as "the presentation of the work of another person as one's own or without proper acknowledgement."

This includes material copied word for word from books, journals, Internet sites, professor's course notes, etc. It refers to material that is paraphrased but closely resembles the original source. It also includes for example the work of a fellow student, an answer on a quiz, data for a lab report, a paper or assignment completed by another student. It might be a paper purchased from any source. Plagiarism does not refer to words alone –it can refer to copying images, graphs, tables and ideas. "Presentation" is not limited to written work. It includes oral presentations, computer assignments and artistic works. Finally, if you translate the work of another person into any other language and do not cite the source, this is also plagiarism.

In Simple Words:

Do not copy, paraphrase or translate anything from anywhere without saying where you obtained it.

Graduate Attributes: N/A

Course Learning Outcomes (CLOs):

By the end of this semester, students are expected to master the following software engineering concepts.

- Software analytics tools
- Software repositories
- Experiential learning, by contributing to an existing open-source project

Health and Safety Guidelines

All health and safety rules specific to this course can be found in the lab manual. General health and safety instructions and available health and safety trainings can be found at:

Safety Programs - Concordia University (https://www.concordia.ca/campus-life/safety/general-safety.html)

On Campus Resources

Please visit Student services at Concordia University for the services available Gina Cody School students.