

COMP 354
Introduction to Software Engineering
Department Of
Computer Science and Software Engineering
FALL 2023

Course Instructor:

Name: Rajagopalan Jayakumar

Email: Jayakumar@cse.concordia.ca

Office Hours:

Office: ER 903

Office Hours: Tuesdays, 15:00-16:00 or by appointment at other times.

Tutorials:

There are three concurrent tutorial sections on Fridays 13:15-14:05. These tutorials will be used to provide instructions and help on the assignments and project. The submitted assignments and project may be evaluated during the tutorials and labs.

Labs:

There are three concurrent lab sections on Fridays 14:20-15:50. These labs will be used to discuss the design and implementation of the software through the assignments and project. The submitted assignments and project may be evaluated during the labs and tutorials.

Course Calendar Description:**Description:**

Software development process models (e.g. linear vs. iterative). Project management; roles, activities and deliverables for each software life cycle phase. Requirements management: analysis, elicitation, and scope. Architecture, design and the mapping of requirements to design and design to implementation. Traceability. Software quality assurance: verification, validation and the role of testing. Maintenance and evolution. Project.

Component(s):

Lecture 3 hours per week; Tutorial 1 hour per week; Laboratory 2 hour per week

Notes:

Students who have received credit for SOEN 341 may not take this course for credit. Students in the BEng in Software Engineering program may not take this course for credit.

Prerequisites

The following courses must be completed previously: COMP 352; ENCS 282.

Specific Knowledge and Skills Needed for this Course:

Students taking this course are expected to have sufficient knowledge of the following topics: discrete mathematics, data structures, algorithms and sufficient programming experience.

Should you have difficulties in any of these topics, you are strongly encouraged to review them before the DNE deadline.

Course materials**Required Textbook:**

Roger Pressman and Bruce Maxim, *Software Engineering: A Practitioner's Approach*, 9th Edition, McGraw Hill, 2020, ISBN13: 9781259872976.

Grading Scheme

Assignments:	20%
Project:	20%
Quizzes:	30%
Final Exam:	30%

There will be three assignments and a project. ***Students must do (and get non-zero marks in) at least two of the three assignments and the project to pass the course.*** The assignments will be on the different phases (requirements gathering, architectural design, and testing) of a simple software system and the project will be to implement the designed software and test it accordingly. Thus, students will learn many of the software development methodologies, tools, and techniques through team assignments. The team project will involve using the methodologies and techniques learned from the assignments to develop a simple software system. ***Assignments and projects must be electronically submitted on the due date and late submissions will not be accepted. All submissions should (would be assumed to) adhere to the ENCS Expectations of Originality.***

There will be five surprise (unannounced) quizzes throughout the term. ***There will not be any make up quizzes, so you will lose the marks for any missed quiz. Students must do (and get non-zero marks in) at least three of the five quizzes to pass the course.***

At the end of the term, there will be a timed final exam on Moodle scheduled by the exam office. The marks from all these quizzes, assignments, project, and the final exam will be used to decide the final grade of a student according to the grading scheme above.

In order to pass the course, a student must achieve an acceptable level of performance (usually, 70% of the class average) both in the practical work (assignments + project) and in the examinations (quizzes + final).

Please note that there is no standard relationship between the numerical percentages and the final letter grades given in this course, except that higher numerical percentages correspond to better grades. The final grade will be based on the total mark in comparison to the class average.

Please note that in the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change.

Tentative Course Schedule

WEEK	TOPIC	MATERIAL
1	Introduction to Software Development	Chapter 1
2–3	The Software Process	Chapters 2–5
3–4	Software Requirements	Chapters 7–8
4–6	Software Architecture and Design	Chapters 9–14
6–7	Software Quality Assurance	Chapters 15–17
8–9	Software Testing	Chapters 19–21
10	Software Configuration Management	Chapter 22
11–12	Software Project and Risk Management	Chapters 24–26

Details on assessment tools:

During this Fall 2023 term, this course will be delivered in person: all components and discussions (lectures, tutorials, labs, demos, quizzes and exams) will be in-person. Lecture notes, assignments, projects and related information will be posted on the Moodle course page as and when they are available. In order to ensure timely learning of the material, there will be quizzes throughout the term on these lecture materials and these quizzes will also be discussed during the class discussions

The practical work/learning will be through three team assignments and a team project. Each team will have about 6 members who should communicate and discuss each of the assignments and project and the submissions should indicate the contributions by each team member and will be evaluated accordingly. The Teaching Assistants (TAs) will provide necessary tutorials for the practical work. After a practical work (assignments and project) is submitted, it will be evaluated by the markers during the tutorial and lab times.

Other information

Students should be aware of the University's Academic Code of Conduct as specified in Section 17.10 in the 2023–2024 Undergraduate Calendar, especially the parts concerning cheating, plagiarism, and the possible consequence of violating this code (available at

<https://www.concordia.ca/academics/undergraduate/calendar/current/section-17-university-secretariat/section-17-10-academic-integrity-and-the-academic-code-of-conduct.html>).

Graduate Attributes:

As this course is on software system design and development, it focuses on the assessment of design attribute (architectural and detailed design). In addition, the following attributes will also be assessed: knowledge base, problem analysis (problem identification and formulation), use of engineering tools (ability to use appropriate tools, techniques and resources) and team work (delivering results).

Course Learning Outcomes (CLOs):

The objective of this course is to understand the software development process and the tools and techniques used to properly design software systems. The course will cover the following.

The Software Process: Process models, agility, human aspects of software engineering.

Software Modeling: Requirements modeling, architectural design, component-level design, user experience design, pattern-based design.

Software Quality: Software quality assurance, software security engineering, software testing, software configuration management, software metrics and analytics.

Managing Software Projects: Project management concepts, creating a viable software plan, risk management.

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 \(https://www.concordia.ca/campus-life/safety/general-safety.html\)](https://www.concordia.ca/campus-life/safety/general-safety.html)

On Campus Resources

Please visit [Student services at Concordia University](#) for the services available to Gina Cody School students.