COMP 3004AB for Winter 2025

Object-Oriented Software Engineering

Course Information

Instructor: Vojislav D Radonjic
Contact: vojislavradonjic AT cmail.carleton.ca
Classroom: see Carleton Central
Lectures: Sec A: Monday and Wednesday 13:05 to 14:25 (in-person)
Sec B: Tuesday and Thursday 10:05 to 11:25 (online via zoom)
Tutorials: this course does not have tutorials
Course Website: available on January 6th

For information about Carleton's academic year, including registration and withdrawal dates, see Carleton's Academic Calendar.

Teaching Assistants

A list of teaching assistants and their contact/office hours information will be posted once the course starts.

Course Calendar Description

Development of object-oriented software systems: theory and practice. Topics include: Computer ethics, software development processes, requirement specification, UML (class, scenario, and state) modeling, object-oriented design patterns, and traceability. Students are to complete a team project.
Includes: Experiential Learning Activity
Precludes additional credit for SYSC 3020, SYSC 3120, SYSC 4120.
Prerequisite(s): COMP 2401 with a minimum grade of C-, (COMP 2404 or SYSC 3010 or SYSC 3110) with a minimum grade of C-, and (COMP 2406 or SYSC 4504).
Lectures three hours a week.

Required Textbook(s) and Other Resources

There is no single textbook. All reading and reference material will be posted on Brightspace.

SCS Laptop Requirement (only applies to on-campus courses)

Every student that has been enrolled in a 1000-level (i.e., first year) course offered by the School of Computer Science after the 2020/2021 school year is required to have a laptop. This includes COMP1001, COMP1005, and COMP1006. For more information, please visit

https://carleton.ca/scs/scs-laptop-requirement/ and then review the requirements at https://carleton.ca/scs/scs-laptop-requirement/laptop-specs/.

Topics Covered and Learning Outcomes

This course teaches an *agile* yet model-driven approach to object-oriented software development (OOSD). More precisely, we will use a ***scenario-driven*** repository-based approach emphasizing testability. The use of ***design patterns*** will guide how we organize object-oriented code for ease of change.

- Understand the advantages of agile development over a waterfall process
- Know the essentials of UML 2.0
- Practice scenario-driven OOSD focusing on its emphasis on traceability from requirements to scenarios to tests.
- Practice software testing
- Gain experience with using a software repository
- Understand a representative subset of the Gang of Four (Go4) object-oriented design patterns
- Gain experience with the team-based development of (models and code for) a medium-sized project
- Gain experience with the use of continuous integration and Go4 patterns.

- Important dates and deadlines can be found here: https://carleton.ca/registrar/registration/dates/academic-dates/, including class suspension for fall, winter breaks, and statutory holidays.

Assessment Scheme

1) 3 *individual* assignments 50%

- ➔ A1 (10%) Developing use cases, due **Januray 22nd.**
- ➔ A2 (15%) Analysis and design a control system, **due February 10th**.
- ➔ A3 (25%) Implementation and testing in Qt/C++ of the system designed in A2, **due March 3rd**.

Collaboration for these assignments is strictly disallowed.

2) *Team* Project: 50% (with **weekly** deliverables)

due April 8th 11:59PM with individual demos starting April 11th

- ➔ As will be explained in class, **continuous integration is essential for success**.
- ➔ You will receive **individual** marks for the project. These marks are based on:
  - your specific contributions
  - your team's weekly progress
  - TA and instructor specific evaluations

    - Individual review of the project with a design and code walkthrough with your assigned TA or the instructor
- ➔ Some time will be devoted in the term to team issues.
- ➔ Collaboration on team projects is restricted to members of the same team. **Inter-team collaboration is strictly disallowed**.
- ➔ A fundamental learning objective of this course is teamwork. *Do not ask to work alone*. Do expect heated discussions if not conflicts in the process of working in a team. Lecture time will be devoted to team issues.

About passing the course

- In order to get a D- or better in this course:
    - The *total* of the marks you accumulate for the 3 individual assignments must be 25 or greater (out of 50)
    - The mark you obtain on the team project must be 25 or greater (out of 50)

Software

All course deliverables are in Qt C++: no other language is allowed. Students will be using Github and the Qt framework on the course virtual machine **COMP3004-F24.ova** (found at https://carleton.ca/scs/tech-support/virtual-machines/). **Students are expected to learn *by themselves* all tools used in this course.**

Undergraduate Academic Advisors

The Undergraduate Advisors for the School of Computer Science are available in Room 5302HP; or by email at scs.ug.advisor@cunet.carleton.ca. The undergraduate advisors can assist with information about prerequisites and preclusions, course substitutions/equivalencies, understanding your academic audit and the remaining requirements for graduation. The undergraduate advisors will also refer students to appropriate resources such as the Science Student Success Centre, Learning Support Services and Writing Tutorial Services.

SCS Computer Laboratory

Students taking a COMP course can access the SCS computer labs. The lab schedule and location can be found at: https://carleton.ca/scs/tech-support/computer-laboratories/. All SCS computer lab and technical support information can be found at: https://carleton.ca/scs/tech-support/. Technical support staff may be contacted in-person or virtually, see this page for details: https://carleton.ca/scs/tech-support/contact-it-support/.

University Policies:

- **Academic Accommodations**

Carleton is committed to providing academic accessibility for all individuals. Please review the academic accommodation available to students here: https://students.carleton.ca/course-outline/.

- **Academic Integrity**

Student Academic Integrity Policy. Every student should be familiar with the Carleton University Student Academic Integrity policy. A student found in violation of academic integrity standards may be sanctioned with penalties which range from a reprimand to receiving a grade of F in the course, or even being suspended or expelled from the University. Examples of punishable offences include plagiarism and unauthorized collaboration. Any such reported offences will be reviewed by the office of the Dean of Science. More information on this policy may be found on the ODS Academic Integrity page: Academic Integrity | Faculty of Science (carleton.ca).

Plagiarism. As defined by Senate, "plagiarism is presenting, whether intentional or not, the ideas, expression of ideas or work of others as one's own". Such reported offences will be reviewed by the office of the Dean of Science. More information and standard sanction guidelines can be found here: https://science.carleton.ca/students/academic-integrity/. Please note that content generated by an unauthorized A.I.-based tool *is* considered plagiarized material.

Unauthorized Collaboration. Senate policy states that "to ensure fairness and equity in assessment of term work, students shall not co-operate or collaborate in the completion of an academic assignment, in whole or in part, when the instructor has indicated that the assignment is to be completed on an individual basis".