

Carleton University
School of Computer Science
COMP 4004A: Software Quality Engineering
Fall 2023

Last Updated August 15th, 2023

Class Logistics

Lectures: **Most lectures are offered synchronously on Wednesdays and Fridays at 1:05pm. Recording of these lectures is NOT guaranteed (i.e., you are expected to attend the lectures). Several lectures will be offered asynchronously (i.e., posted for you to listen to at your convenience shortly thereafter)**

Fall Break: week of October 23rd (no office/TA hours that week)

Note: **** December 8th follows a Monday schedule ****

Withdrawal by November 15th 2023

‘Office Hours’ via zoom, **by email appointment only**

Instructor Info

Jean-Pierre Corriveau

email: jeanpier AT scs.carleton.ca

No textbook is used in this course

Material for all lectures (including any recordings) will be posted to Brightspace. Brightspace will also be used for announcements and for all student assignment submissions. The 2 exams (midterm and final) will be scheduled **out of lecture times** by exam services. One will be the week of November 10th, one during the exam period. **All students are responsible for immediately gaining access to Brightspace.**

TAs: names, emails and office hours

will be announced in Brightspace

Learning Objectives:

THEORY: (verified via 2 exams)

- Understand the foundations of software testing and software metrics.
- Understand the principles and limitations of test-driven development (TDD) in the context of Agile Development, as well as issues pertaining to refactoring.
- Understand behavior-driven development (BDD) and acceptance testing.
- Understand the principles and limitations of model-based testing (MBT).
- Familiarize yourself with the state of the art in software testing.

PRACTICE: (verified via 3 assignments)

- **Gain experience at learning tools, coding and testing BY YOURSELF.**
- Gain experience with using JUnit and TDD.
- Gain experience with BDD and scenario-based testing using Cucumber.
- Gain experience with testing a web application.

Detailed Course Description

It is STILL widely acknowledged that *software quality* is of the utmost importance. Yet, despite recent advancements in program verification, automatic debugging, assertion deduction and model-based testing (MBT), Ralph Johnson (of Gang of Four design patterns fame) and many others still view software verification as a "catastrophic computer science failure".

In this course, we first and foremost explore the issue of software **testing**, that is, the **execution** of software in order to find errors. A first pervasive concern will be *test automation*, which is necessary if testing is to be objective, systematic, and scalable. A second pervasive concern will be the *testing of scenarios* (e.g., as captured in use cases, user stories, use case maps).

After an introduction to the theoretical foundations of validation and verification, we will consider a code-oriented approach to testing, focusing, in particular, on test driven development (TDD) and the strengths and limitations of **JUnit** and unit testing. We will then address acceptance testing and focus on model-based techniques and the use of **Cucumber** for it. We will also include a discussion of *smells* and refactoring and their relevance to software testing. We will also address software metrics, and of the state-of-the-art in software testing.

Prerequisites

A student registering in COMP 4004 is assumed to have a **strong** background in object-oriented design **and** programming. More specifically, a student should have taken COMP 2401, 2406 *and* 3004, or their equivalent.

You are expected to work individually and autonomously (i.e. learning by yourself).

Attendance and Exams

You are expected to attend the synchronous online lectures AND to study the material posted online.

More specific information about what will be addressed in the midterm (i.e., the **in-term out-of-lecture-time**) exam will be confirmed in class. That exam will be a **closed-book** exam and will involve the pencil and paper development of test suites. It is to take place **tentatively** November 10th, 11th or 12th. **The exact date, time and location will be announced in class and in Brightspace once finalized by exam services.**

The final exam, scheduled during the exam period of the Fall term, will be an open-book (but no computer) multiple-choice exam pertaining to most of the material presented in class.

Software

For the assignments, students will be using a **PRIVATE** repository (Github), a dependency manager (Maven), an IDE (IntelliJ) and Java, as well as JUnit, Cucumber and Selenium. Exact details are to follow in Brightspace.

Most importantly, students are expected to learn *by themselves* all tools used in this course. Beyond the presentations on September 15th and November 10th, lectures will NOT be discussing the technical aspects of the installation and use of the tools required by the assignments. Questions on these issues should be directed to the TA Andrew Hlynka (email in Brightspace) and it must be emphasized that it is a key learning objective to have students tackle mostly on their own such issues (especially given that, in the past, most installation problems stemmed from specificities of the laptop of a student, NOT the tool to install).

There are several tutorials available online for the tools we use; in particular for JUnit (<http://www.vogella.com/tutorials/JUnit/article.html>) and Cucumber.

Suggested *optional* readings

- 1) Robert Binder (good introduction to OO Testing)
Object-oriented Testing, Addison-Wesley
- 2) K. Naik and P. Tripathy (pricey but excellent reference for s/w testing)
Software Testing and Quality Assurance, Wiley, 2008

Evaluation scheme (no SAT/UNS marking)

3 INDIVIDUAL Testing Assignments 18% + 15% + 17%
• each with coding & actual practice of a s/w testing approach

2 EXAMS (1 midterm and 1 final) 25% each

A1: TDD of the Tournament game due: October 13th
• use of a repository, Maven and JUnit

A2: Acceptance testing for this game due: November 10th
• use of Cucumber

A3: Coding and testing the web app version of this game due: December 8th
• use of a web driver (Selenium)

Logistics

Late submissions WILL NOT BE ACCEPTED and will get a mark of 0.

Assignment submission is handled electronically and there is no "grace period" with respect to a deadline. Technical problems do not exempt you from this requirement, so if you wait until the last minute and then have issues with your connection, you will still receive a mark of zero. **Do not expect extensions for the assignments.**

Furthermore

- 1) There will be NO supplemental or grade raising exam in this course.
- 2) **No mark or extra work can be substituted for another mark.**
- 3) Should you be sick during an assignment, you must submit the appropriate form to the instructor BEFORE the assignment is due.
You *may* get a new deadline for the current assignment BUT this will NOT grant you more time for the subsequent assignments.
- 4) **The Fall term ends December 8th and no submission is accepted after that date.**
- 5) **Your combined mark for the two exams (each worth 25% of the final grade) must be greater than 24 out of 50 in order to pass the course.**
- 6) For each assignment, you will be submitting one or more files that contain source code, and these files must be given the correct filename and be provided in the specified format. Assignments that are incorrectly named or in the incorrect format will be penalized and may receive a mark of zero.
- 7) Assignments are to be done **individually** *without any sharing*. Collaboration between students is *not* permitted: all alleged plagiarism will be reported to the office of the dean of Science (ODS). Penalties for such offences can be found on the ODS webpage: <https://science.carleton.ca/academic-integrity/>.
In particular, posting *any* work put in a public location (eg a GitHub **public** repository, or Chegg, CourseHero, OneClass) constitutes a form of sharing that enables plagiarism. As such, if your work is plagiarized from a public posting of yours, you will be considered to have participated in this offence!

All material created for this course (including exams and assignments) remains the exclusive intellectual property of their author(s). They are intended for personal use and may not be reproduced or redistributed without prior written consent of the author(s).

Undergraduate Academic Advisor

The undergraduate advisor for the School of Computer Science is available in Room 5302C HP, by telephone at 520-2600, ext. 4364 or by email at undergraduate_advisor@scs.carleton.ca. The advisor can assist with information about prerequisites and preclusions, course substitutions/equivalencies, understanding your academic audit and the remaining requirements for graduation. The undergraduate advisor will also refer students to appropriate resources such as the Science Student Success Centre, Learning Support Services and the Writing Tutorial Services.

Science Student Success Centre (SSSC)

The Science Student Success Centre is a central advising unit for students in Science courses. We help students achieve their goals by providing access to resources, workshops and activities that enhance their academic and study skills, and help them make key connections with their peers. Mentors can help you customize an individual study plan which includes weekly and semester work or study schedules, and also help when you need information on developing a new study strategy, obtaining summer job opportunities, or clarifying ideas and concepts to better understand and cope with new course content. Science mentors can help you **learn how to learn what you need to learn** for your classes.

Drop by the Science Student Success Centre at 3431 Herzberg Laboratories or visit

<http://sssc.carleton.ca/>

University Policies

For information about Carleton's academic year, including registration and withdrawal dates, see <https://calendar.carleton.ca/academicyear/>

Pregnancy Obligation. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, visit:

<https://carleton.ca/womensstudies/resources-and-links/equity-services/>

Religious Obligation. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details:

<https://carleton.ca/equity/focus/discrimination-harassment/religious-spiritual-observances/>.

Academic Accommodations for Students with Disabilities If you have a documented disability requiring academic accommodations in this course, please contact the Paul Menton Centre for Students with Disabilities (PMC) at 613-520-6608 or pmc@carleton.ca for a formal evaluation or contact your PMC coordinator to send your instructor your Letter of Accommodation at the beginning of the term. You must also contact the PMC no later than two weeks before the first in-class scheduled test or exam requiring accommodation (if applicable). After requesting accommodation from PMC, meet with your instructor as soon as possible to ensure accommodation arrangements are made. For more details, visit the [Paul Menton Centre](#) website.

Survivors of Sexual Violence. As a community, Carleton University is committed to maintaining a positive learning, working and living environment where sexual violence will not be tolerated, and survivors are supported through academic accommodations as per Carleton's Sexual Violence Policy. For more information about the services available at the university and to obtain information about sexual violence and/or support, visit: carleton.ca/sexual-violence-support

Accommodation for Student Activities. Carleton University recognizes the substantial benefits, both to the individual student and for the university, that result from a student participating in activities beyond the classroom experience. Reasonable accommodation must be provided to students who compete or perform at the national or international level. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, see [the policy](#).

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 awarded penalties that range from a reprimand to receiving a grade of *F* in the course or even being expelled from the program or University. Examples of punishable offences include: plagiarism and unauthorized co-operation or collaboration. Information on this policy may be found [here](#).

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.

Unauthorized Co-operation or 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". Please refer to the course outline statement or the instructor concerning this issue.

Medical Certificate

The following is a link to the official medical certificate accepted by Carleton University for the deferral of final examinations or assignments in undergraduate courses. To access the form, please go to <https://carleton.ca/registrar/cu-files/medical-certificate-form/>
