

Carleton University
School of Computer Science
COMP 4004A: Software Quality Engineering
COMP 5104: Object-Oriented Software Development
Fall 2021
Last Updated September 14th, 2021

Class Logistics

- Lectures: **offered online synchronously**
(i.e., via Zoom at the lecture times posted in Carleton central)
- Fall Break: week of October 25 (no office/TA hours that week)
- ‘Office Hours’ held ONLINE starting the week of September 13th and run until Dec 9th
October 11th is a holiday.
**** December 10th follows a Monday schedule ****

Instructor Info

Jean-Pierre Corriveau
email: jeanpier AT scs.carleton.ca
office hours: 1-2 pm Monday and Wednesdays (via Zoom)

Course Website

<http://people.scs.carleton.ca/~jeanpier//Fall2021/>

No textbook is used in this course

Some material will be posted to Brightspace, and possibly some to the course’s website. Brightspace will also be used for announcements and for assignment submission.
uOttawa students are responsible for immediately gaining access to Brightspace

TAs: names, emails and office hours

will be announced in Brightspace

Learning Objectives

- **Gain experience at learning tools, coding and testing BY YOURSELF**
- Understand the fundamental principles of agile development
- Understand the principles and limitations of TDD
- Gain experience with using JUnit and TDD
- Understand behavior-driven development (BDD) and scenario-based testing
- Gain experience with BDD, scenario-based testing and using Cucumber
- Gain experience with testing a web application
- Gain experience with testing advanced characteristics of code (e.g., UI, concurrency)

Time permitting, for COMP 5104: (and those students of 4004 wanting to review this material):

- Review a representative subset of the Gang of Four (Go4) design patterns
- Review the essentials of UML 2.0

Detailed Course Description

It is 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* (as captured in use cases, user stories, use case maps, and/or message sequence charts).

We will start with a more code-oriented (industry-relevant) approach to testing, focusing in particular on test driven development (TDD) and the strengths and limitations of **JUnit** and unit testing. In this context, we will discuss the challenges of *refactoring* (production *and* test) code and the usefulness of unit test patterns and code metrics.

We will also consider scenario modeling, acceptance testing and Cucumber.

Other topics include:

- state-of-the-art *model-based* approaches to software testing
- some of the theoretical foundations of validation and verification

Prerequisites

A student registering in COMP 4004 or COMP 5104 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. Because there are no exams, the assignments of COMP 4004 and COMP 5104 are **time-consuming** and presuppose you **already** know how to develop:

- a) a client/server application
- b) a web application

These topics will NOT be reviewed in this course.

Additionally, familiarity with UI development and/or the programming of asynchronous applications constitutes an asset for the last assignment of this course.

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

Attendance

You are expected to attend most lectures of this course.

**** Your Zoom login ID must clearly identify you using your first and last name****

While I will *try* to record and post lectures, it must be emphasized that:

- you should NOT plan on *systematically* missing online lectures
- there is always a risk the recording of a lecture is incomplete or absent due to unforeseen technical difficulties.

Software

For the assignments, students will be using a repository (e.g., Github), a dependency manager (e.g., Maven), an IDE (e.g., IntelliJ) and Java as well as Junit, Cucumber and other testing tools. Exact details are to follow in Brightspace.

Most importantly, students are expected to learn *by themselves* all tools used in this course. That is, 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 TAs and it must be emphasized that it is a key learning objective to have students tackle mostly on their own such issues. 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 2000
- 2) K. Naik and P. Tripathy (pricey but excellent reference for s/w testing)
Software Testing and Quality Assurance, Wiley, 2008

Evaluation scheme

4 INDIVIDUAL Assignments **25% each**

- | | |
|---|--------------------|
| A1: TDD of a networked <i>card game</i> for 2 to 4 players
• use of a repository and of JUnit | due: October 1st |
| A2: BDD of strategies for this game
• use of Cucumber | due: October 22nd |
| A3: Running this game through browsers
• use of a web driver (e.g., Selenium) | due: November 19th |
| A4: Advanced testing
• Develop and test a real-time version of this game or of a simpler one
• OR develop and test a graphical user interface for this game or for a simpler one
• Each one of these 4 different options requires you to first research which testing tool(s) to use and then learn and use your choice of tool(s).
• The maximum mark achievable for developing and testing a real-time version of or a GUI for the simpler game will be lower than the one for tackling the game of the previous assignments. | due: December 10th |

Logistics

Late assignment 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.

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. Each assignment is worth 25% of your final grade and no 'weight transfer' can be done from one assignment to another.
- 3) Should you be sick for an assignment, you must submit the appropriate form to the instructor BEFORE the assignment is due. Your physician must state when you are expected to be cured and, if possible, how many days you were incapacitated. Should your medical certificate/form be accepted, you will get a new deadline for the current assignment BUT this will NOT grant you more time for the subsequent assignments.
- 4) **You must pass, that is, obtain a grade greater than 49% on EACH assignment in order to pass this course.**
- 5) 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.
- 6) 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 assignments) remain 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 [Carleton's Academic Calendar](#).

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 [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/>
