

Course Syllabus



Last updated: Saturday, September 7, 2024

[tentative until this line is removed]

CSC207H1: Software Design

Calendar description:

An introduction to software design and development concepts, methods, and tools using a statically-typed object-oriented programming language such as Java. Topics from: version control, unit testing, refactoring, object-oriented design and development, design patterns, advanced IDE usage, regular expressions, and reflection.

Learning Objectives:

By the end of this course, you will:

- be able to effectively communicate about software design with your peers
- be comfortable using version control
- be able to write Java code to satisfy program specifications
- be able to apply the SOLID design principles, design patterns, and Clean Architecture to design object-oriented software solutions
- have experienced what it is like to work in a collaborative software development environment
- have a deeper understanding of how to approach testing your code
- be confident in your ability to use more advanced features of your IDE

Course Contact Information

Course email address: csc207-2024-09@cs.toronto.edu (<mailto:csc207-2024-09@cs.toronto.edu>)

Course coordinator: Jonathan Calver (BA4222)

Lab coordinator: Sophia Huynh

Instructional support: Amin Gillani

Lecture Schedule

Lecture times, locations, and instructors for each section

| Lecture Section | Lecture Time | Location | Instructor |
|-----------------|---------------|----------|-------------------|
| LEC 0101 | T, Th 10–11am | EM001 | Jonathan Calver |
| LEC 0201 | T, Th 1–2pm | EM001 | Paul Gries |
| LEC 0301 | T, Th 2–3pm | UC140 | Yasaman Rohanifar |
| LEC 0401 | T, Th 3–4pm | RW117 | Jonathan Calver |
| LEC 0501 | T, Th 4–5pm | BA1130 | Lindsey Shorser |
| LEC 5101 | Th 6–8pm | KP108 | Lindsey Shorser |

Office Hours

Instructor and TA office hours will be held weekly throughout the term. The schedule will be posted on the [Office Hours \(https://q.utoronto.ca/courses/353377/pages/office-hours\)](https://q.utoronto.ca/courses/353377/pages/office-hours) page. **We encourage you to also make use of Piazza and the weekly tutorial time with your peers and TAs to ask questions as you learn the course material.**

Course Project

A major component of your experience in this course is contributing to a group software design project (teams of 4–5 students). For the first half of the term, we will be meeting during the weekly Monday tutorial time slots to work on skill development activities in a group setting. In the last half of the term, you will apply these skills in a software development project, which will see you and your team build an application by applying what you have been learning in this course.

Final Exam

The final exam is a comprehensive exam. **You need to achieve at least 40% on the final exam; otherwise, your course grade will be no higher than 47% and you cannot pass the course.**

Marking Scheme

Course marking scheme

| Course Work Title | Portion of Course Mark | Due Date / Notes |
|-------------------------------|-------------------------------|---|
| Weekly Quercus Review Quizzes | 5% (0.5% each; best 10 of 11) | Due Fridays at 6pm during the term: <ul style="list-style-type: none"> September 13, 20, 27 October 4, 11, 18, 25 November 8, 15, 22, 29 |
| 5 Assignments | 20% (4% each) | Due Fridays at 6pm during the first half of the term: <ul style="list-style-type: none"> September 13, 20, 27 October 11, 25 |
| Midterm Test | 10% | During the tutorial on October 7th <ul style="list-style-type: none"> 50-minutes within each TUT section, half the class will write in the first hour and the other half in the second hour |
| 2 Ethics Research Surveys | 1% (.5% each) | "Pre-module" survey released September 23, 5pm and due September 30, 11:59pm "Post-module" survey released November 29, 5pm and due December 2, 11:59pm |
| Group Project | 25% | Presentations on Monday / Tuesday (MakeUp Monday) of last week of classes during your tutorial time. Grade based on final presentation (15%) and your overall contribution to the project (10%). |
| Final Exam | 39% | To be scheduled by the Faculty of Arts and Science |

Course Topics by Block

The course is divided into three 4-week blocks.

Block 1: Software Developer Skills and Tools

- Version Control
- Java OOP
- Testing
- Refactoring

Block 2: Principles of Software Design

- SOLID design principles
- Clean Architecture (CA)
- Design patterns

Block 3: Professional and Miscellaneous Topics

- ethics modules on users and accessible design
- regular expressions (regex)
- communication and code review
- GenAI and prompt engineering
- interviewing for software jobs

Tutorials

All room information for the weekly tutorials will be posted in the CSC207H1 F Tutorial (All Sections) Quercus instance. Tutorials take place on Mondays in the Bahen computer labs during your registered TUT section. Information about how to request section swaps to accommodate scheduling conflicts will be announced on Quercus during the term.


Textbook

Most of the core design concepts discussed in this course can be found in [Clean Architecture](#) by Robert Martin. The textbook is optional, but highly recommended. Past students have shared that they found the textbook to be very useful. You may also find the following optional books to be interesting reads:

- [Object-Oriented Design & Patterns](#) by Cay Horstmann is quite good; the first chapter is a nice crash course on Java
- [Effective Java](#) by Joshua Bloch (highly recommended if you plan to code more in Java beyond this course; its emphasis is on how you can best use Java — it doesn't teach the syntax)
- [Program Development in Java](#) by Barbara Liskov with John Guttag (takes a very formal approach to software design; in particular, their UI–FP (user interface – functional part) partitioning of the design of a system and their subsequent discussion fits well with our discussion of Clean Architecture in this course)

- Clean Code and Clean Craftsmanship by Robert Martin are also worth reading if you enjoy the author's writing style in Clean Architecture
- Refactoring by Martin Fowler is a great reference for the refactoring topic

Piazza

We will be using [Piazza for class discussion](https://piazza.com/class/m0b6ljczidm1gs)  [_ \(https://piazza.com/class/m0b6ljczidm1gs\)_](https://piazza.com/class/m0b6ljczidm1gs). Once enrolled in the course, you should be automatically added to Piazza. The system is highly catered to getting you help fast and efficiently from both classmates and instructors. We encourage you to post course content questions on Piazza — don't be shy! If you have any problems or feedback for the developers of Piazza, you can email [team@piazza.com \(mailto:team@piazza.com\)](mailto:team@piazza.com). From experience, they are very responsive and even open to implementing requested features!

Late Policy

Late Policy for Individual Work

You are responsible for meeting all deadlines. All individual assignments will be submitted on MarkUs at 6pm on Fridays during the term. To provide reasonable flexibility with your schedule, you may use grace tokens on MarkUs to extend the deadline without penalty up to ~~48~~ 72 hours. For each deadline on MarkUs, you will receive 6 grace credits each worth 12 hours. There will be no extensions on the weekly Quercus review quizzes though. Homework will not be accepted beyond that time without special consideration (see below).

If you're experiencing illness, mental health crises, family/personal emergencies, or other exceptional circumstances beyond your control that prevent you from being able to complete an assessment on time, you can apply for special consideration by completing a form. This form will be posted on the course Quercus page ahead of the first assignment.

Please note that special consideration cannot be granted to accommodate for heavy course load, multiple assignments and/or tests scheduled during the same period, or challenges with time management. Grace tokens are intended to be used in these situations instead.

In the case of illness, please email your completed form to the course address as soon as possible and we'll make appropriate accommodations, for example, re-weighting of missed term work to the final exam.

Late Policy for Group Work

The course project requires group work. If your group experiences delays due to illness of one or more of your group members, please discuss with your tutorial TA or contact the course address to discuss reasonable accommodations.

Remark Requests

All remark requests will be done through MarkUs, and they will be handled before final course grades are submitted. If requesting a remark, you must submit such requests within two weeks of the work being returned.

Accessibility

If you have an acute or ongoing disability issue or accommodation need, you should register with Accessibility Services (AS) at the beginning of the academic year by visiting <http://www.studentlife.utoronto.ca/as/new-registration> (<http://www.studentlife.utoronto.ca/as/new-registration>). Without registration, you will not be able to verify your situation with your instructors. AS will assess your situation, develop an accommodation plan with you, and support you in requesting accommodation for your course work. Remember that the process of accommodation is private: AS will not share details of your needs or condition with any instructor, and your instructors will not reveal that you are registered with AS. Please reach out to the course address if you have any questions or concerns.

Academic Integrity

Please familiarize yourself with the Rules and Regulations from the U of T Calendar (especially the Code of Behaviour on Academic Matters): <http://www.artsci.utoronto.ca/osai> (<http://www.artsci.utoronto.ca/osai>)

Use of Generative AI

The use of Generative AI is allowed throughout the course. With this said, we caution you to not rely entirely on these tools to complete your coursework. Instead, we recommend treating the use of generative AI as a supplementary tool only for exploration and engaging with the course material. Ultimately, you (and not any AI tool) are responsible for your own learning in this course, and for all the work you submit for credit. It is your responsibility to critically evaluate the content generated, and to regularly assess your own learning independent of generative AI tools. Over reliance on generative AI may give you a false sense of how much you've actually learned, which can lead to poor performance on the final exam, in later courses, or in future work or studies after graduation.