Course Syllabus

✎ Edit

Last updated: Wednesday, August 29, 2025

```
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: csc207-2025-09@cs.toronto.edu (mailto:csc207-2025-09@cs.toronto.edu)

Course coordinator: Jonathan Calver (BA4222)

Lab coordinator: Sophia Huynh

Instructional support: Cindy Tran

Lecture Schedule

Lecture Section	Lecture Time	Location	Instructor
LEC 0101	W 1–3pm	BR120	Jonathan Calver
LEC 0201	W 3–5pm	MP102	Lindsey Shorser
LEC 0301/2	Th 1–3pm	MY150	Paul Gries / Jonathan Calver
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 **Lectures, Labs, and Office Hours (https://q.utoronto.ca/courses/394773/pages/lectures-labs-and-office-hours)** page.

Team-Based Learning (TBL)

This course follows a TBL approach. You will be working with a course-assigned team of 5–6 students starting in the second week of the term. This will be your team throughout the entire course — including the group project. Details about team formation will be provided during the first week of the course. If you enrol late, please email the course address to ensure that you get added to a team in a timely manner.

Course Project

A major component of this course is getting to experience contributing to a group software design project. For the first half of the term, we will be meeting during the weekly tutorial time slots to work on skill development activities in a group setting, and begin the planning stage of your group project. In the last half of the term, you will apply these skills in a software development project, which will see

you and your team collaboratively build a Java desktop application.

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
iRATs	5% (1.25% each; best 4 of 5)	individual readiness assurance tests **You will need to bring a device (e.g., cellphone, laptop, or tablet) to complete the quiz online through Quercus and TeamUp!** There will be a practice test at the beginning of lecture during the week of September 8th. These will take place at the beginning of lectures on the following weeks: • September 15, 29 • October 13 • November 3, 17
tRATs	5% (1.25% each; best 4 of 5)	team readiness assurance tests These will each take place in class immediately after each iRAT is completed.
Module Mark	15% (best 4 of 5)	• Each of modules 1–5 will have graded work. Details will be provided per module.
Out of Class Midterm Test	10%	• Friday, October 3rd, 5:15–6:45pm (90-minutes)

2 Ethics Research Surveys	1% (.5% each)	• surveys will be announced when available • first survey due by September 24, 1pm • second survey due by end of December 2
Group Project	20%	Presentations during last lab of the term. Grade based on final presentation (10%) and your overall contribution to the project based on git logs and an individual written report (10%).
Peer Evaluations	5%	These will be interpreted by your lab TA and instructors to determine your grade.
Final Exam	39%	To be scheduled by the Faculty of Arts and Science

---

Course Topics by Module

The course is divided into six 2-week modules.

Module 0: Java Fundamentals and Version Control

Module 1: Java OOP

Module 2: Design Principles

Module 3: Clean Architecture

Module 4: Design Patterns

Module 5: Ethics, Regex, and Presentations

Tutorials

Tutorials will begin in the second week of the course (the week of September 8th).

All room information for the weekly tutorials will be posted on Quercus. Tutorials take place on Mondays/Tuesdays in the Bahen computer labs during your registered TUT section. TUT sections and LEC sections are linked, as you will be working with your team in lectures and in labs.

Textbook

Course notes are available **on GitHub (https://github.com/CSC207-UofT/207-course-notes)** and will be updated during the term. We recommend the **official Java tutorials (https://docs.oracle.com/javase/tutorial/)** as a reference as we learn Java during the first few weeks of this course.

Most of the core design concepts discussed in this course can be found in Clean Architecture by Robert Martin. The book is optional, but highly recommended. Past students have shared that they found the book 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 code refactoring

---

Piazza

We will be using Piazza for class discussion. 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)**. From experience, they are very responsive and even open to implementing requested features!

Late Policy

Late Policy for Individual Work

To provide reasonable flexibility with your schedule, you may use a grace token on MarkUs to extend any deadline for individual work without penalty by up to 72 hours. You will receive 1 grace credit

worth 72 hours per deadline in the course (so you can use an extension on every MarkUs homework if you wish). 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 the form below:

Special Consideration Form (link to be added once term begins)

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.

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.

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 on these tools to complete your work. Instead, we recommend using such tools to engage with the course material as you learn. Ultimately, you 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.
