

# CSC301

---

Introduction to Software Engineering

# What is CSC301 about?

## Developing software

- In a realistic environment
- With users in mind
- With a clear purpose and value
- On an on-going basis, in a traceable manner
- As part of a team
- While dealing with changing requirements

# Course Goals

- Improve coding skills
- Introduce software development life-cycle
  - Much more than just coding
- Get you to think as pragmatic professionals
  - Tooling as part of your workflow
  - Articulate goals, define success metrics and make data-driven decisions
  - Identify common problems/challenges, and apply well-known, generic solutions

# Topics

- Software Tools
  - Version control (Git)
  - Project management (GitHub)
  - Build and/or automation (Travis CI, Maven)
  - IDE and/or debugger (Eclipse)
- A pragmatic approach to tooling
  - Professionals use tools to be more efficient
  - And build custom tools (focusing on “bang for the buck”), when they are needed.

# Topics

- Project management
  - Software processes
  - Focus on modern Agile techniques
  - Ex: Scrum, Kanban and Test-Driven Development
- A pragmatic approach to team organization
  - Collaboration comes with an overhead
  - Organize a team's workflow  $\Leftrightarrow$  Reduce overhead

# Topics

- Product management
  - Articulating *what* we're building, *who* we're building it for and *why* it is useful/valuable.
  - Standard planning tools/techniques such as personas, user stories, diagrams, mock-ups, etc.
  - Scoping and defining a Minimum Viable Product.
- A pragmatic approach to product decisions
  - Define success metrics → Experiment & collect user feedback → Make data-driven decision(s)
  - Do it frequently and incrementally improve your product

# Topics

- Software design & Coding

- Best practices and common pitfalls
- Design patterns such as Iterator, Adapter, Observer/Observable, Abstract Factory and Builder.
- Code craftsmanship
- Various useful topics in software engineering

Ex: Lambda expressions (aka callbacks), serialization and persistence, asynchronous programming, lazy-loading and caching, distributed applications, etc.

Logistics ...

# Resources

- Course website:

<http://csc301-fall-2016.github.io>

- Discussion Board:

<https://piazza.com/utoronto.ca/fall2016/csc301>

- GitHub organization:

<https://github.com/csc301-fall-2016>

# Instructor

- Joey Freund
- Email: [joey@cs.toronto.edu](mailto:joey@cs.toronto.edu)
  - If your question is of general interest to the class, please consider posting it on the discussion board, instead of sending an email.
- Office: BA5244
- Office hours: TBD

# Lectures & Tutorials

- Lectures

- L0101 (day section) : Monday 12-2 @ BA1200
- L5101 (evening section) : Monday 6-8 @ BA1200

- Tutorials

- L0101 (day section) : Tuesday 12-1 @ BA1200, LM157 & RW142
- L5101 (evening section) : Monday 8-9 @ BA1200, BA1230 & BA1240

# Prerequisites

- CSC209
  - Implicit prerequisite, CSC207.
  - Basic Object-Oriented programming in Java
  - Comfortable with Unix command line
- CSC263/5
  - Basic data structures and types.
  - Ex: Array, List, Queue, Stack, Map (aka dictionary), Tree, Graph.

# Marking Scheme

4 Individual Programming Assignments	35%
Individual Final Demo	15%
Term Test	15%
Team Project	35%

No Final exam!

# Individual Assignments

- 4 Java coding assignments
- Specified as JUnit tests
- Auto-marked
- Focus:
  - Reading and writing object-oriented code
  - Hands on experience with professional tools  
Ex: Git, GitHub, Travis CI and Maven.
  - Applying design patterns

# Individual Final Demo

- One-on-one with a TA
- 10-15 minute demo + Q&A
  - Presenting your individual work from the previous few weeks.
- During the week November 28
- Focus:
  - Software design & implementation
  - Testing
  - Good use of standard tools

# Team Project

- ~8 weeks long
- 5-7 students per team
- One TA per team, acting as a “mentor”.
- Focus:
  - Product definition & building a prototype
  - Process & team organization
  - Presenting your work

# Team Project

## ● Worth 35% of your final grade:

- 15% - Two deliverables, worth 7.5% each
- 10% - Final demo
  - During the last week of the term
- 10% - Consistent individual contribution
  - Commit history & graphs on GitHub
  - Participation in tutorials throughout the term

# Cheating

- Don't cheat!
- Feel free to discuss ideas with others, but don't take notes or share code with others.
- When in doubt, ask your instructor or TA.

Please keep in mind that CSC301 is a hands-on course!

In other words - A lot of fun, but also a lot of work.