

# COMP 250

Introduction to Computer Science

Fall 2017

Instructor: Prof. Michael Langer

My “personal” web page: <http://www.cim.mcgill.ca/~langer/>



## Michael Langer

I am an Associate Professor in the [School of Computer Science](#) at [McGill University](#) in Montreal, Canada. I'm also a member of CIM ([Center for Intelligent Machines](#)).

I grew up in Toronto. I received a B.Sc. (Math) from McGill in 1986, M.Sc. in Computer Science from the [University of Toronto](#) in 1988, and Ph.D. from McGill in 1994. I was a post-doc at the [NEC Research Institute](#) in Princeton NJ, and at the [Max-Planck-Institute for Biological Cybernetics](#) in Tuebingen Germany where I was a [Humboldt Research Fellow](#). I returned to McGill as a faculty member in 2000. Here is my [Curriculum Vita](#).

### Contact

office: McConnell Eng. 329  
postal: McConnell Eng. #318  
3480 University  
Montreal H3A0E9  
QC, Canada  
phone: 514-398-3740  
fax: 514-398-3883  
email: [langer\[ @ \]cim.mcgill.ca](mailto:langer[@]cim.mcgill.ca)

### Research

Publications  
[PDF links](#)  
[Google Scholar](#)  
[DBLP](#)  
[ResearchGate](#)

[Talks and Posters](#)

### Teaching

COMP 250 [Introduction to Computer Science](#)  
COMP 251 [Algorithms and Data Structures](#)  
COMP 273 [Introduction to Computer Systems](#)  
COMP 423 [Data Compression](#)  
COMP 546 [Computational Perception](#)  
COMP 557 [Fundamentals of Computer Graphics](#)  
COMP 558 [Fundamentals of Computer Vision](#)

### Links

[Grad studies in CS](#)  
[PhD comics](#)

# My public COMP 250 web page

<p>← → ↻ 🏠 ⓘ www.cim.mcgill.ca/~langer/250.html ☆ 🔔 ⚙️ S 📄 S</p> <p>📱 Apps ★ Bookmarks 📁 Bookmarks ☁️ Dropbox   📁 Other bookmarks</p> <p><b>Office hours :</b> MWR 13:00-14:00 ENGMC 329 <b>Lectures:</b> MWF 10:35-11:25 MCMED 522 (recorded) ← ? MRF 14:35-15:25 MD-HAR <del>310</del> G-10</p>	
<b>Resources (New)</b> <ul style="list-style-type: none"><li>• <a href="#">Course Outline (Fall 2017)</a></li></ul>	<b>Resources (Old)</b> <ul style="list-style-type: none"><li>• free Java book: <a href="#">"How to think like a computer scientist"</a></li><li>• Eclipse <a href="#">Download</a></li></ul>
<b>LECTURES</b>	<b>EXERCISES, ASSIGNMENTS</b>
<b>Preliminaries</b> <ol style="list-style-type: none"><li>0. introductions <a href="#">(slides)</a></li><li>1. grade school algorithms for arithmetic <a href="#">(notes)</a> <a href="#">(slides)</a></li><li>2. binary numbers <a href="#">(notes)</a> <a href="#">(slides)</a></li><li>3. Eclipse IDE and debugging (in class tutorial)</li></ol>	Exercises 1: number representations <a href="#">(PDF )</a>
<b>Linear Data Structures</b> <ol style="list-style-type: none"><li>4. array list <a href="#">(notes)</a> <a href="#">(slides)</a></li><li>5. singly linked lists <a href="#">(notes)</a> <a href="#">(slides)</a></li><li>6. doubly linked lists <a href="#">(notes)</a> <a href="#">(slides)</a></li><li>7. list sorting algorithms <a href="#">(notes)</a> <a href="#">(slides)</a> <i>bubble, selection, &amp; insertion sort</i></li><li>8. stacks <a href="#">(notes)</a> <a href="#">(slides)</a></li><li>9. queues <a href="#">(notes)</a> <a href="#">(slides)</a></li></ol>	Exercises 2: arrays <a href="#">(PDF )</a> Exercises 3: linked lists <a href="#">(PDF )</a> <a href="#">(code)</a> <a href="#">E3 Q2 figures</a> Exercises 4: $O(n^2)$ sorting <a href="#">(PDF )</a> Exercises 5: stacks and queues <a href="#">(PDF )</a>

# Welcome !

There are 650 of you enrolled in the two sections.

Sec. 001: 450 chairs in McMED 522 (MWF 10:35-11:25)

Sec. 002: 190 chairs in MDHAR 310 (MRF 14:35-15:25)

I will treat the two sections as one course.

(More on this issue later...)

# Who are you (degree) ?

B. Arts	110
B. Arts & Sci	15
B. Com	25
B. Sci	260
B. Soft. Eng.	35
B. Eng	<u>175</u>
	650

# Who are you (year) ?

U0	25
U1 new	80
U1 returning	170
U2	230
U3	130
other	<u>5</u>
	640

# Who are you (year) ?

U0	25	
U1 new	80	
U1 returning	170	
U2	230	← switch
U3	130	← \$
other	<u>5</u>	
	640	

# COMP 250: overview

- Preliminaries  
(grade school algorithms, binary numbers, Eclipse)
- Data Structures 1  
(lists: array lists, linked lists, stacks and queues)



# COMP 250: overview

- Preliminaries

(grade school algorithms, binary numbers, Eclipse)

- Data Structures 1

(lists: array lists, linked lists, stacks and queues)

- Induction and Recursion

(recursive algorithms for lists e.g. searching and sorting)

- Mathematical Tools for Analysis of Algorithms

(recurrences, asymptotic complexity and 'big O')

# COMP 250: overview

- Preliminaries  
(grade school algorithms, binary numbers, Eclipse)
- Data Structures 1  
(lists: array lists, linked lists, stacks and queues)
- Induction and Recursion  
(recursive algorithms for lists e.g. searching and sorting)
- Mathematical Tools for Analysis of Algorithms  
(recurrences, asymptotic complexity and 'big O')
- Data Structures 2  
(trees, binary search trees, heaps, graphs, hash tables)
- Object Oriented Design in Java  
(inheritance, polymorphism)

<https://www.mcgill.ca/study/2017-2018/courses/comp-250>

# COMP 250 Introduction to Computer Science (3 credits)

Offered by: Computer Science ([Faculty of Science](#))

## Overview

Computer Science (Sci) : Mathematical tools (binary numbers, induction, recurrence relations, asymptotic complexity, establishing correctness of programs), Data structures (arrays, stacks, queues, linked lists, trees, binary trees, binary search trees, heaps, hash tables), Recursive and non-recursive algorithms (searching and sorting, tree and graph traversal). Abstract data types, inheritance. Selected topics.

**Terms: Fall 2017, Winter 2018**

Instructors: Michael Langer (Fall) Jérôme Waldispühl (Winter)

- 3 hours
- Prerequisites: Familiarity with a high level programming language and CEGEP level Math.
- Students with limited programming experience should take [COMP 202](#) or equivalent before [COMP 250](#). See [COMP 202](#) Course Description for a list of topics.






**COMP 250 uses the Java programming language.**

# What is your programming background?

256 attempts have been completed

## Question 1

Which of the following applies to you? (If more than one applies, then indicate the first.)

I have taken and passed COMP 202.		170	(66.41 %)
I have taken a different Java course (high school or CEGEP) or learned Java on my own, and I have spent over 40 hours coding in Java.		46	(17.97 %)
I have taken a course in C programming (e.g. COMP 208) and I have spent over 40 hours coding in C.		17	(6.64 %)
I have learned one of Python/Javascript/Matlab/R and I have spent over 40 hours coding in it.		19	(7.42 %)
None of the above. (In this case, you should take COMP 202 instead of COMP 250, or you should email me or see me in person and explain your special circumstances.)		4	(1.56 %)

So, 85% of you know how to code in Java  
(COMP 202 or equivalent -- ECSE 202 is fine too)

## COMP 202 Foundations of Programming

Offered by: Computer Science ([Faculty of Science](#))

### Overview

Computer Science (Sci) : Introduction to computer programming in a high level language: variables, expressions, primitive types, methods, conditionals, loops. Introduction to algorithms, data structures (arrays, strings), modular software design, libraries, file input/output, debugging, exception handling. Selected topics.

- Prerequisite: a CEGEP level mathematics course
- Restrictions: [COMP 202](#) and [COMP 208](#) cannot both be taken for credit. [COMP 202](#) is intended as a general introductory course, while [COMP 208](#) is intended for students interested in scientific computation. [COMP 202](#) cannot be taken for credit with or after [COMP 250](#)

6% of you know how to code in C (or C++)  
(COMP 208)

## COMP 208 Computers in Engineering

Offered by: Computer Science ([Faculty of Engineering](#))

### Overview

Computer Science (Sci) : Introduction to computer systems. Concepts and structures for high level programming. Elements of structured programming using FORTRAN 90 and C. Numerical algorithms such as root finding, numerical integration and differential equations. Non-numerical algorithms for sorting and searching.

You have a lot work to do in next two weeks: you need to learn about *objects and classes* in Java.

(C++ is object oriented, so you are in better shape.)

7% of you know how to code in Python, Javascript, R, Matlab, ... but not Java or C

You need to spend 40 (?) hours in the next 2 weeks learning Java:

- types (primitive vs. reference)
- objects and classes

Otherwise, (1) take COMP 202 instead or (2) spend some time in Fall learning Java on your own. Then take COMP 250 in Winter.

# How to learn Java in 2 weeks?

Resources suggested on the Course Outline:

- See COMP 202 content (topics, slides)  
<http://www.cs.mcgill.ca/~cs202/2016-01/web/sec2.html>
- online tutorials
- free book: “How to think like a computer scientist”
- books from library

**Don't just read. Code! Do exercises.**



# Math Prerequisites

CEGEP level math (Cal 1)

$$1 + 2 + 3 + 4 + \dots \quad n = \frac{n(n+1)}{2}$$

$$1 + x + x^2 + x^3 + \dots \quad x^n = \frac{1 - x^{n+1}}{1 - x}$$

$$\log ab = \log a + \log b$$

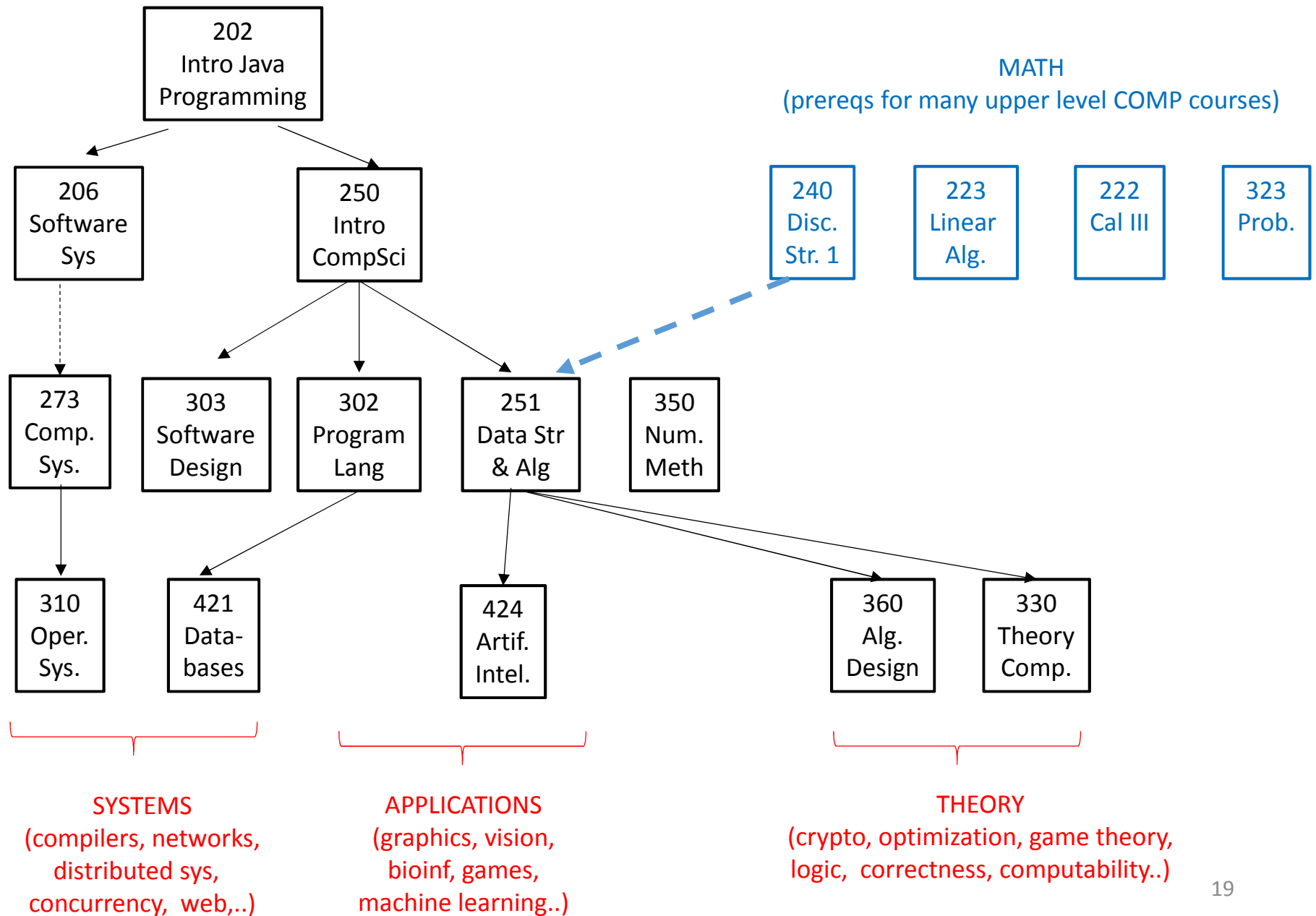
# Unofficial “coreq” MATH courses

CS Major/Minor  
B. Eng & B. SoftEng } MATH 240 Discrete Structures 1

Joint Math/CS programs: MATH 235 Algebra I

One of these courses is the official *co-requisite* for  
COMP 251 Algorithms and Data Structures.  
I advise you to it *before COMP 251*.

# The Big Picture: COMP@McGill



COMP 250 + COMP 206 + COMP 273 ?

Only if you have a lot of programming  
experience already.

If you take just two, do 250 and 206.

# COMP 250 Resources

## public web page

<http://www.cim.mcgill.ca/~langer/250.html>

- Course Outline
- lecture notes
- slides
- exercises
- .....

## MyCourses/250

- announcements  
*(you need to subscribe)*
- lecture recordings
- discussion boards
- submit assignments
- quizzes
- grade book

# Office hours

- MWR 1-2 pm (Mon, Wed, Thurs)
- by appointment (send me email)
- quick questions by email  
(if question is of general interest then I will ask you to post on mycourses discussion board)

# CSUS Helpdesk

HOURS: 12pm - 5pm (Mon-Fri)  
LOCATION: Trottier 3090

## WHO ARE WE? WHAT DO WE DO?

- U2 and U3 students who have taken this course and want to help you!
  - We are a **FREE** drop-in tutoring service, perfect for study help, and guidance on assignments.
  - We provide review sessions for midterms and finals for intro courses!
-

# Evaluation (your grade)

- 4 x 10% Assignments

Java programming

- 10 %      Six Quizzes (***on mycourses, not during class***)

Details/dates to be announced (first time I do this...)

- 50 %    Final Exam    (multiple choice)



# Assignments:

collaboration vs. cheating

- ✓ clarification Q's on mycourses Discussion Board
- ✓ hands on helping your peers to debug code
- x posting key ideas on mycourses (or facebook)
- x sharing code

# Quizzes

On mycourses, multiple choice, graded automatically.

Do them *on your own*.

They are worth only 10% of your grade.

They are mainly there as:

- an incentive for you to keep up with the course
- a way for you to gauge how you are doing
- a way to practice for the final exam

# How much work can you expect?

Faculty of Science suggests a 3 credit should require 9 hours work per week:

$$\begin{aligned} & 13 \text{ weeks} * 9 \text{ hours/credit/week} \\ & = 117 \text{ hours total} \end{aligned}$$

So, 5 courses implies ~45 hours per week.

4 courses implies ~36 hours per week.

# How much work can you expect ?

- Lectures (~40 hours)
- Quiz/Exam study, Exercises (~40 hours)
- Assignments (~40 hours = 4 x 10)

*That is my guesstimate for a B grade for a B student.*

# Two Sections or One ?

Sec. 001: 450 seats in McMED 522 (MWF 10:35-11:25)

Sec. 002: 190 seats in MDHAR 310 (MRF 14:35-15:25)

Lectures will be recorded.

I will treat the two sections as one course.

If the *majority* of you choose not to attend the lectures, then I *will* use one section (probably Sec 002 MRF 14:35-15:25) for tutorials rather than give an identical lecture.

# COMP 206 and the 15 minute uphill walk for Sec. 001

COMP 206:

Sec. 001: ADAMS AUD (MWF 9:35-10:25) → ...

Sec. 002: MAAS 10 (MWF 10:35-11:25)

COMP 250:

... → Sec. 001: McMED 522 (MWF 10:35-11:25)

Sec. 002: MDHAR 310 (MRF 14:35-15:25)

# Your TODO list

- Review/learn Java (COMP 202)

Assignment 1 will be posted in ~2 weeks.

- Learn how to use IDE for Java e.g. Eclipse

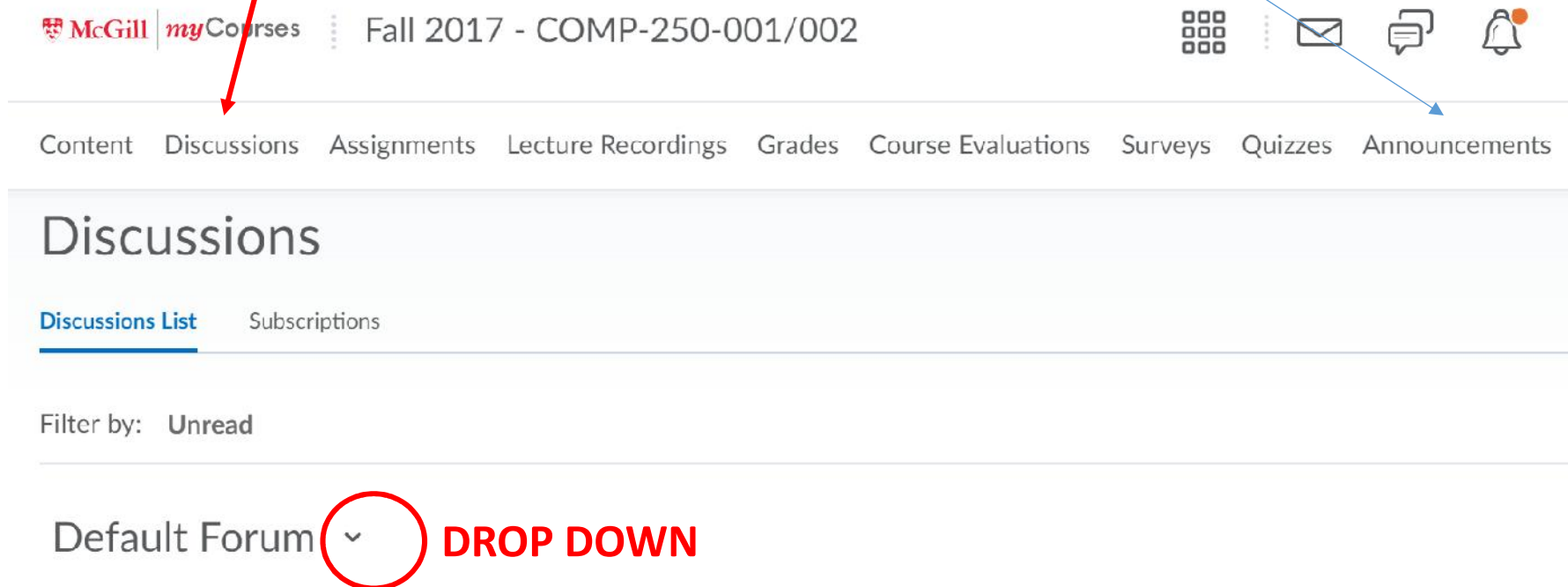
(or NetBeans or IntelliJ or some other, but not DrJava)

In class tutorial for Eclipse next week (probably Wed/Thurs)

**INSTALL and run Eclipse beforehand** <https://www.eclipse.org/home/>

- Subscribe to mycourses: Announcements, Discussions.

# Subscribe to Announcements and Discussion Board




McGill myCourses Fall 2017 - COMP-250-001/002

Content Discussions Assignments Lecture Recordings Grades Course Evaluations Surveys Quizzes Announcements

## Discussions

[Discussions List](#) Subscriptions

Filter by: Unread

Default Forum  **DROP DOWN**