CSC236 Fall 2016

Introduction to the Theory of Computation

Instructor

Larry Zhang

Email: ylzhang@cs.toronto.edu

Office: DH-3076

TAs

Alex Kornilenko, Anwar Abu Ghosh, Chris Chianelli, Chris Koehler, David Zolnieryk, Mansi Vaghela, and Peggy Cao

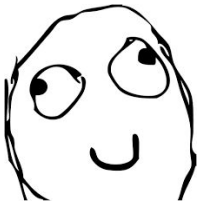
Outline for today

- **Why** take CSC236
- **What** is in CSC236
- **How** to do well in CSC236

Why take CSC236?

The difference between

an average programmer who can code stuff



and

a university-trained **computer scientist** who can understand and design elegant and efficient programs

You CANNOT call yourself a “computer science person” before taking CSC236.

What is in CSC236

It is your first **theory** course in CS

- We won't do much coding like in CSC108/148.
- Computer science is not about how to code
  - like astronomy is not about how to use telescopes
- CS is about problem-solving.
  - Understand the "physics" of problems
  - Create solutions utilising the special tool (computers).

We’re going to study

- Reasoning about **program correctness**
  - so you know your program will be correct before you code it
- Reasoning about **program efficiency**
  - so you know how fast your program will run before you code it
- **Designing** algorithms
  - Some tricks for designing quick algorithms, just a start

More concretely

These are the **fundamentals** of any interesting stuff you might want to do with computers.

- **Induction (simple, complete, structural)**
  - The technique for "**reasoning**"
- **Asymptotic Notations (Big Oh/Omega/Theta)**
  - Describe program runtimes.
- **Recursion**
  - Recursion in CSC148 was magic; now you can explain the magic.
  - We will learn how to analyse and design recursive programs.
- **Program correctness, invariant**
  - Formal way to prove a (iterative/recursive) program is correct
- **Regular expressions and finite automata**
  - Models for a computer's understanding of **languages**.

How to do well in CSC236

First of all ...

Be interested.

Course web page

`http://www.cs.toronto.edu/~ylzhang/csc236/`

All course materials can be found on the course web page.

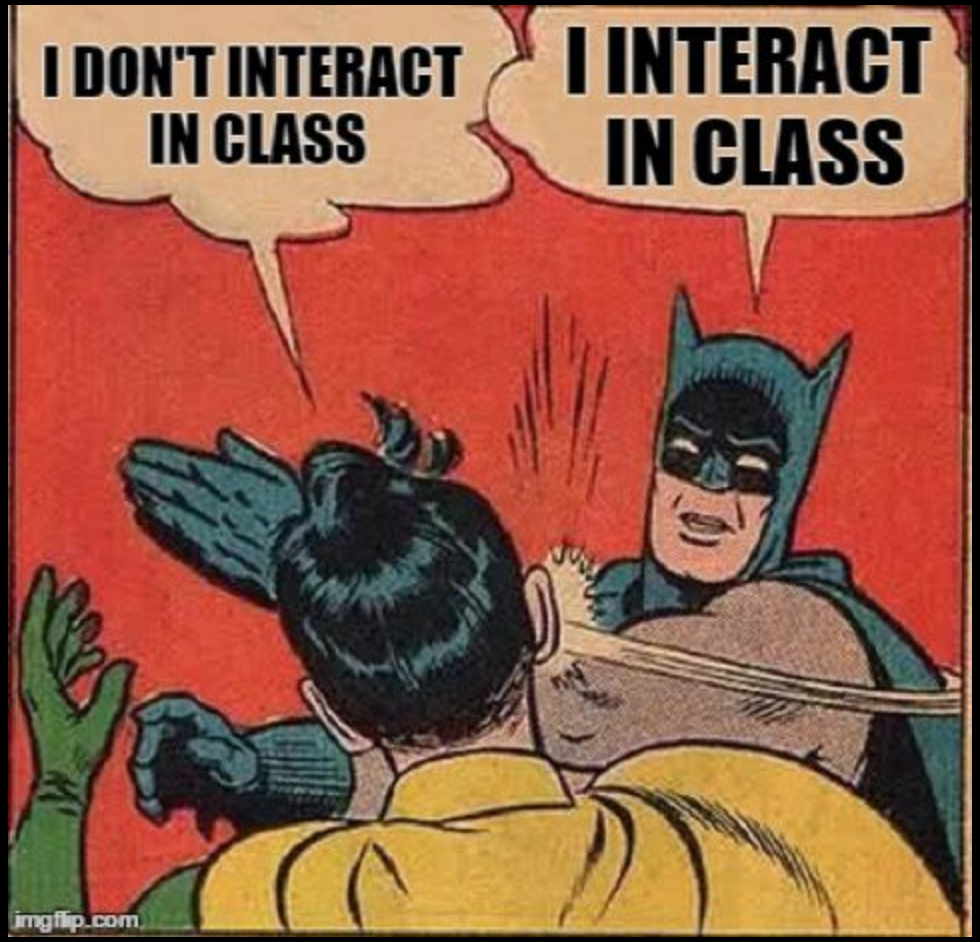
Textbooks

- **CSC236 Course Notes**: written by David Liu, edited by Dan Zingaro (available online for free)
- **Invariants: a Generative Approach to Programming**: written by Dan Zingaro (available online for free)
- Reading links for each week on the course web page.

Lectures

- L0101: Monday 1-3pm, at DV-2082
- L0102 Wednesday 3-5pm, at IB-345
- Get involved in classroom **interactions**
  - ➔ answering a question
  - ➔ making a guess / bet / vote
  - ➔ back-of-the envelope calculations (**bring pen&paper!**)

Emotional involvement makes the brain remember better!


I DON'T INTERACT IN CLASS
I INTERACT IN CLASS
imgflip.com

Tutorials

- Unlike in lectures, you will be in “**active learning mode**” in the tutorials.
- You’ll try to solve a problem yourself using what’s just learned from the lecture.

TUT0101: Wednesday 3pm - 4pm, at DV-1154

TUT0102: Wednesday 4pm - 5pm, at DV-1154

TUT0103: Thursday 9am - 10am, at DV-1154

TUT0104: Thursday 10am - 11am, at DV-1154

TUT0105: Friday 9am - 10am, at DV-1154

TUT0106: Friday 10am - 11am, at DV-1154

TUT0107: Friday 9am - 10am, at DV-1151

TUT0108: Friday 10am - 11am, at DV-1151

Tutorials are as important as lectures!

- Tutorial notes will be posted shortly before the tutorial, so you may work on it beforehand if you want.
- No posted solutions for tutorials, you need to go there.
- You should NOT skip tutorials.
- “I'll just skip the gym for this week and work out at home instead” -- this almost never works.

Practice is the key to the success in CSC236.

Problem Sets (9 of them)

- Weekly practice, except for weeks with tests
- 9 sets in total
- You may work in groups of 1-3
- **Collaborate intelligently!**
  - Rule of thumb: Collaborate in such a way that **everyone in the group** can pass the final exam
  - Groups can be different for different PS.

Typing your problem sets

- **Submission must be typed. Use LaTeX**
- The best way to type formulas
- TeX source files for PS will be posted, which you can use as templates.
- Check the course web page under “Homeworks” for LaTeX resources.
- Handy tools that let you do everything in the browser
  - www.sharelatex.com

Marking Scheme

- **Problem sets: 30%**
  - Your lowest mark for PS will be dropped
- **Two midterm tests: 30%**
  - Better one counts for 18%, the other for 12%
- **Final Exam: 40%**
  - You must get at least 40% of the final exam to pass the course.

Bonus!

2% to the final mark, for active participation in classroom interactions and Piazza discussions.

- 2%: very active participation
- 1%: active participation
- Good practices:
  - Active interaction in lectures
  - Ask good questions and give good answers to questions on Piazza

Piazza (Discussion Board)

https://piazza.com/utoronto.ca/fall2016/csc236h5

For all course related discussions.

All announcements will be posted on Piazza.

Daily reading is required.

Protip: Turn off email notifications in the settings, so you don't get annoyed by lots of emails, and when you receive an email you know it's an important announcement from the instructor.

Office Hours

- Monday 3-5pm
- Wednesday 5-7pm
- Or by appointment

Feel free to just drop by and say hi.
If you've never been to an office hour, try it and you won't regret.


GODIVA

Weekly Feedback Form

https://goo.gl/forms/wrNz4woSPqJI4pO72

good feedback = improved learning experience

timely feedback = timely improvement

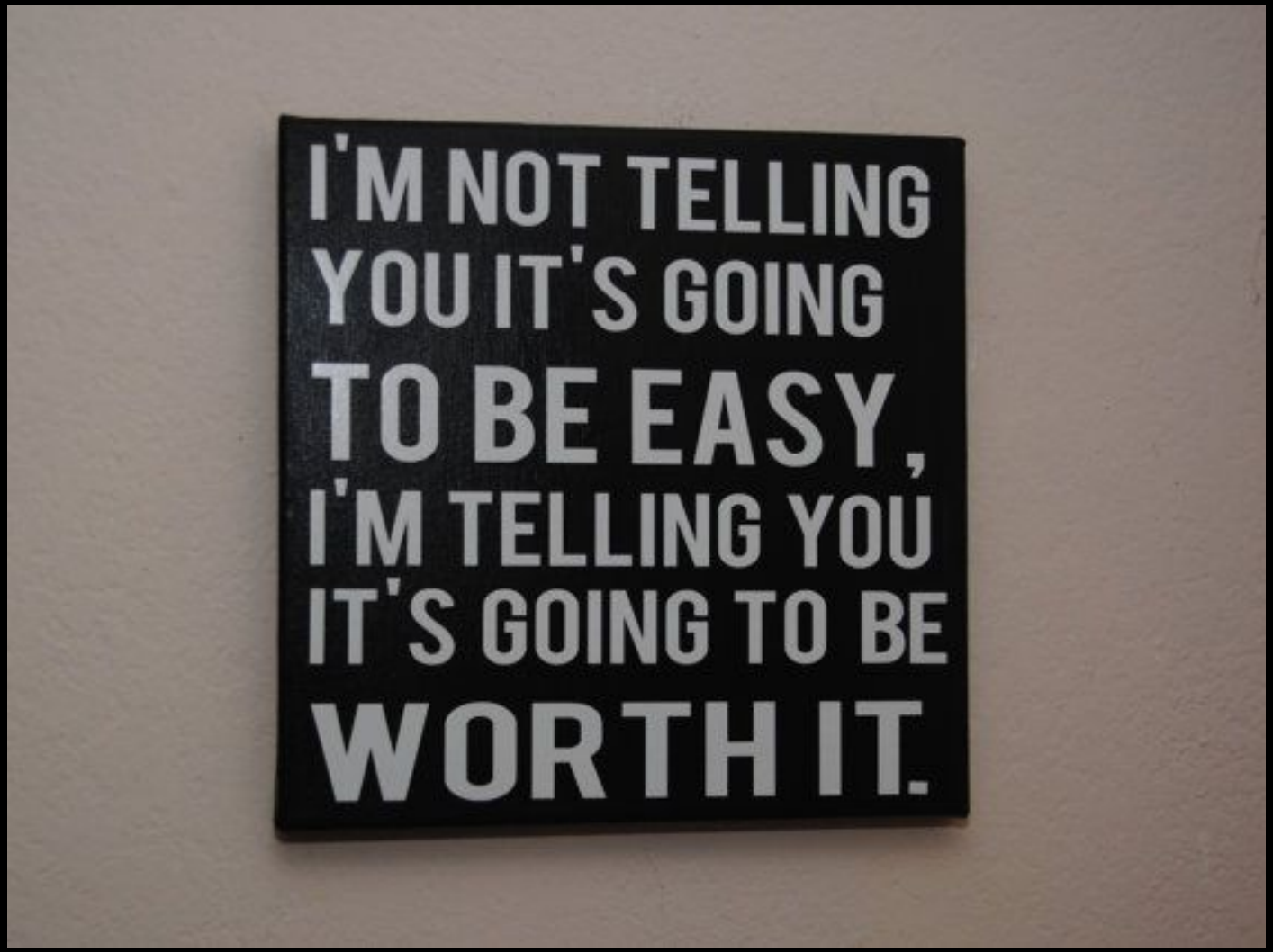
The more, the better!

Academic Integrity

- We are very serious about it.
- It’s not as easy as “don’t cheat”.
- You need to understand what is plagiarism.
- You also need to protect yourself.
- Read the following link:
  - http://www.cs.toronto.edu/~fpitt/documents/plagiarism.html

Checklist: How to do well

- Be interested.
- Check course web page and Piazza daily
- Attend lectures
- Actively solve problems in tutorials
- Read the course notes.
- Discuss on Piazza.
- Go to office hours.
- Work hard on homeworks, and submit on time.
- Give feedbacks weekly
- Do well in tests and exams.
- Don’t cheat.


I'M NOT TELLING
YOU IT'S GOING
TO BE EASY,
I'M TELLING YOU
IT'S GOING TO BE
WORTH IT.