Administrative Issues

EECS2030: Advanced
Object Oriented Programming
Fall 2017

CHEN-WEI WANG

Instructor

- How may you call me?

  "Jackie" (most preferred),

  "Sir", "Professor", "Professor Wang", "Professor Jackie", "Hey", "Hi", "Hello"

- Office: Lassonde Building 2043

- Office hours: 1:30pm – 3:30pm on Mondays, Tuesdays, and Thursdays. Or by appointments.

- When you need advice on the course, speak to me!

- Throughout the semester, feel free to suggest ways to helping your learning.

Course Information

- Course moddle page:
  https://learn.lassonde.yorku.ca
  Log in and look for Section A of EECS2030
- Lecture notes and supplementary materials will be posted on my website:
  http://www.eecs.yorku.ca/course_archive/
  2017-18/F/2030A/
- Check these two sites regularly for announcements and learning resources:
  - notes   [typically posted after the class]
  - lecture recordings and source codes
  - tutorial videos
- Check your emails regularly!

Lectures

- *One person talks at a time* in class, please!
- Use of mobile is *distracting* and even *disrespectful* to me.
- In class: core concepts, examples, *your engagement*
- I tend to do lots of additional illustrations on the board.
- I am experimenting on recording the slide shows, illustration, and voice. But technology may fail sometimes.

  ∴ Invest yourself a *paper* notebook and pencils/pens/erasers!
- Don't be lazy!
  - Take (even incomplete) notes during lectures.
  - Taking a photo of the board/screen is fine, but maintaining hand-written notes will keep you motivated.
- I *can't slow down too much* in the class, for otherwise I can't (nearly) finish the agenda set up by the CS department.
- I will try to *pause* frequently for questions.
- Bring your own laptop, with Eclipse installed.

Attendance

- Attendance to *all* lectures and labs are necessary.
  - We can't check attendance for a large class (after all, it's really your responsibility, no?)
  - Don't consider coming to classes as pressure.
  - It's for your best interests of learning!

- Computer test, based on lab exercises and lecture materials
- Each section has its own lab tests.
- A guide will be available to you about a week before the lab test.

Academic Integrity

The moral code or ethical policy of academia:

- avoidance of cheating or plagiarism;
- maintenance of academic standards;
- honesty and rigor in research and academic publishing.

Pay careful attention to *all* occasions where the submitted work is to be graded and receive credits (i.e., labs, quizzes, assignments, tests, exams).

It is *absolutely not* acceptable if, in any of these occasions, you:

- share your (programming or written) solutions with others;
- copy and paste solutions from elsewhere and claim that they are yours.

What is this course about?

- *Solve problems* .
  - Come up with software artifacts whose structure corresponds to the real life entities. [object orientation]
  - *Step-by-step* and *precise* instructions, by which the computer follows to achieve a certain task. [procedural programming]
- *Express solutions in Java* .

Adapting Yourself to the Second Year

- You had lots of fun in your first-year courses:
  - Programming problems were explained to you via explicit changes on physical devices (e.g., Android tablet).
- However, this may not be how you work as a software developer in reality:
  - Programming problems were explained to you via the expected methods' signatures (input and output types) and a few use cases.
- Being able to think *abstractly* without seeing changes on a physical device is an important skill you are expected to acquire when graduating.
- Think of programming interviews at Google or Microsoft: you are given problems described in English, then you are expected to solve it on a whiteboard.

- Please approach me (email, in person) as soon as possible, so we can make proper arrangements for you.
- We will work out a way for you to gain the most out of this course!

Index (1)