Administrative Issues

EECS2030 B: Advanced
Object Oriented Programming
Fall 2019

CHEN-WEI WANG

Instructor

- How may you call me?

  “Jackie” (most preferred),

  “Professor Jackie”, “Professor”, “Professor Wang”, “Sir”, “Hey”, “Hi”, “Hello”
- Office: Lassonde Building 2043
- Office hours: ***4pm – 6pm*** on ***Mondays***, ***Tuesdays***, and ***Wednesdays***. 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

- Lecture materials will be posted on my website:
  `https://www.eecs.yorku.ca/~jackie/teaching/lectures#EECS2030_F19`
- Two moddle sites: `http://moodle.info.yorku.ca/`
  - `EECS 2030 Fall 2019-2020`
    - Announcement for **all** sections.
    - **Lab instructions** are posted here.
  - `LE/EECS2030 B - Advanced Object Oriented Programming (Fall 2019-2020)`
    - Announcement for Section B only.
    - Post your questions here in the **forum**.
    - Never share solutions to graded components on the forum!!!
- Check your emails regularly!

If You Are Not Enrolled Yet

- Send me an email ASAP requesting access to the course moodle, with your *name*, *student number*, *York Passport ID*.
- Still attend lectures.
- Still complete labs (no extension).

Class Protocol

- No talking, no mobile – ***distracting***, ***disrespectful*** to everyone.
- If you feel like talking or using mobile, please ***leave***.

- In class: core concepts, examples, *your engagement*
- You'd study the ***remaining*** slides/notes on your own.
- Speak to me *early* when you have trouble studying!

Study Tips

I attempt to record each lecture entirely:

- *Not meant to be a replacement for classes!*
- The purpose of recording is that you can focus on reaching ***maximum comprehension***.
  - *Ask questions!*
  - Take (even ***incomplete***) notes: they help when re-visiting lectures.

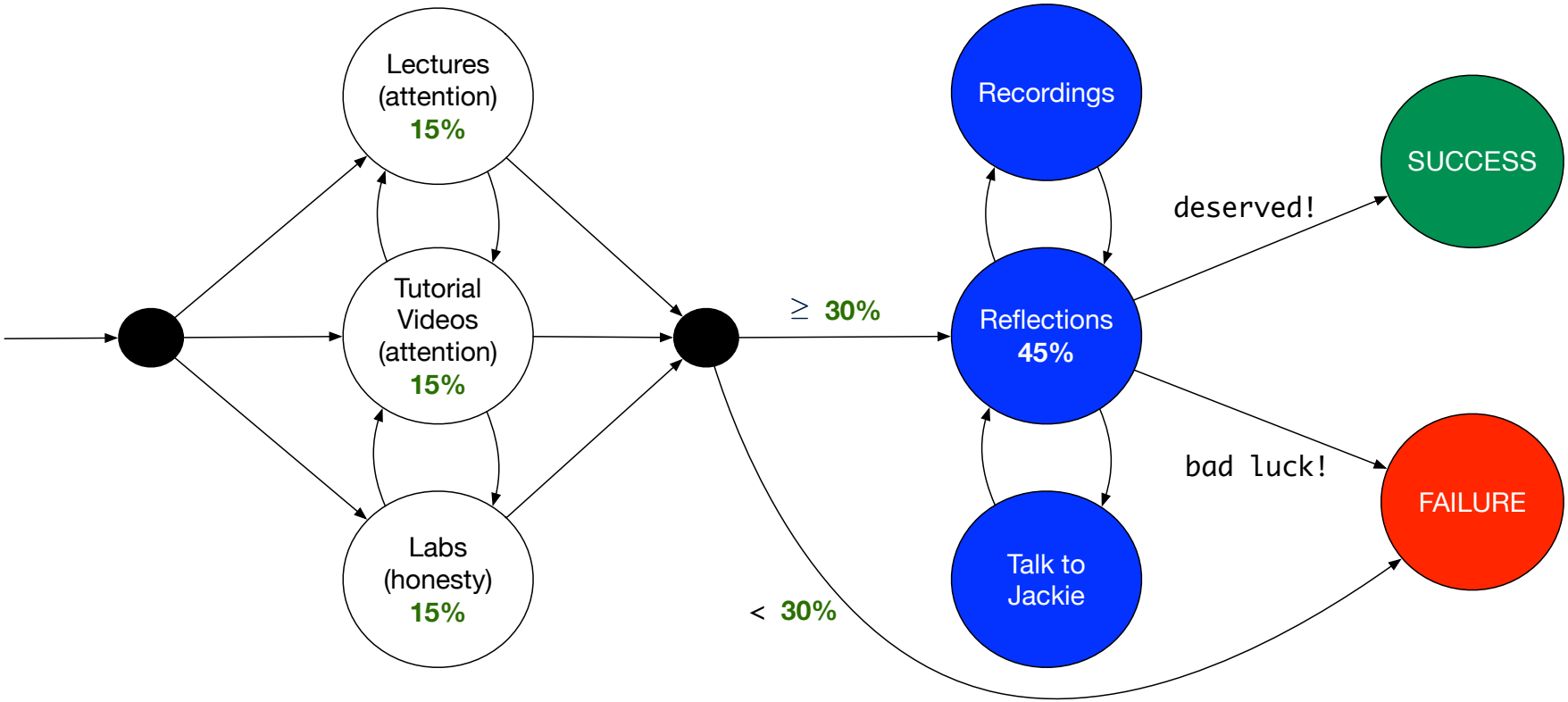
General Tips about Studying CS

- To do well, ***inspiration*** is more important than ***perspiration***.
- Hard work does not necessarily guarantee ***success***, but no success is possible without ***hard work***

  $\Rightarrow$

  - Don't be too satisfied just by the fact that you work hard.
  - Make sure you work hard both on ***mastering "ground stuffs"*** and, more importantly, on ***staying on top of what's being taught***.
  - Always *reflect* yourself on *how things are connected*.
    - Be *curious* about going beyond lectures (e.g., CodingBat).
    - Be *curious* about why things work the way they do.

Survival Pattern of this Course



Lab Tests

- Computer test, based on lab exercises and lecture materials
- Each section has its own lab tests.
- A ***guide*** will be available prior to 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.

Course Syllabus

Available on the Moodle site for Section B.

Lab Sessions

- Lab 0 has been posted on the course Moodle.
  - You must complete Lab 0 from the Prism lab (LAS1006) computers
  - Submissions must be completed using the command line.
- I will attempt to come for all lab sessions.
- Feel free to ask me other course-related materials.

Adapting Yourself to the Second Year

- You had lots of fun in your first-year courses:
  - Programming solutions were developed and tested via ***visualization*** on physical devices (e.g., Android tablet).
  - You may have done a bit of *testing* :
    using a ***Tester class*** with the `main` method.
- However, this isn't how a real ***software developer*** works:
  - Programming ***problems*** are explained via the expected methods' ***headers*** (input and output types) and some ***use cases***, without visualization!
  - A set of ***tests*** must be *re-run automatically* upon changes.
- Thinking *abstractly* without seeing changes on a physical device is an important skill to acquire when graduating.

  e.g., Watch *interviews at Google*: Given problems described in English, solve it on a whiteboard.

What is this course about?

- *Solve problems*.
  - ***Object Orientation***: Come up with software artifacts whose ***architecture*** corresponds to the real life entities.
  - ***Procedural Programming***: **Step-by-step** instructions, by which the computer follows to achieve a certain task.
- *Express solutions in Java*.

Need Accommodation for Tests/Exams?

- 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)