CMPT 280

Intermediate Data Structures and Algorithms
Introduction

Mark G. Eramian

University of Saskatchewan

Course Website

- CMPT 280 course on Moodle: `http://moodle.cs.usask.ca`
- Features:
  - Download lecture notes, assignments, *some* tutorial content.
  - Assignment submission and grades.
  - Discussion forums.
- Students are responsible for reading class announcements daily. Email copies of course announcements will be sent to your University of Saskatchewan email. If you use a different address, forward accordingly.

Main Topics

- Container classes, review of lists (linked and arrayed)
- Cursors and Iterators
- Regression Testing
- Timing Analysis
- Abstract Data Types
- Trees (linked and arrayed)
- Dispensers (stacks, queues, heaps)
- Searchable Dispensers (BSTs, AVL trees)
- Dictionaries (hash tables, 2-3 trees, B-trees, B+ trees)
- Specialized trees (k-D trees, trie trees)
- Graph data structures and algorithms
- Sorting Algorithms (quick, merge, heap, bucket, radix)

Evaluation

- Assignments 40%
- Midterm Exam 15%
- Final Exam 45%

A student must write the final examination to pass the course.

In the case where a student has a passing percentile grade calculated as outlined above, but obtained less than 50% on the weighted combination of the Midterm and Final Exam grades (i.e. less than 30 of the 60 possible percentage points for exams), the student will receive a final grade no higher than 55%.

Midterm Exam

- Confirmed for Wednesday, February 13, 6pm-7:30pm, room TBA, **outside class time** (this is also in the syllabus).
- Please make work/travel arrangements accordingly and ASAP.

Textbook and References

- **Required Text:** M. Eramian, Intermediate Data Structures and Algorithms (Course Readings for CMPT 280), Fourth Edition, 2018. Electronic copies available for free on course website.

Assignment Submissions

- Assignments part written, part programming.
- Electronic submission for **everything** via Moodle.
- Acceptable formats: PDF, DOC, DOCX, RTF, TXT

Tutorials

- Tutorials will be mostly weekly.
- First tutorial: January 16/17.
- **Some** (not all!) tutorial content available online.
- Focus on practical concepts directly relevant to completion of assignments.
- Time allotted to ask for extra help on any part of the course.
- Tutorials are mandatory.
- Tutorial content is fair game for examinations, whether posted online or not.
- Attendance will improve success on assignments and examinations.

Getting Help

- Ask questions in class.
- Ask questions of your tutorial leader.
- Ask questions during Spinks help-desk hours (these hours begin the week of January 21).
- Ask questions in Moodle discussions.
- Email help: when requesting email help, address email to both me (eramian@cs.usask.ca) **and** Peggy Anderson (peggy.anderson@usask.ca). Whoever can respond first will.

When should you ask for Help?

- When you are stuck on the same bug or problem for more than 30 minutes.
- When you aren't sure if your program is producing correct results.
- When it seems like you have to write an absurdly large amount of code.
- When you don't understand the algorithm and/or data structure you are being asked to implement. *Nobody can implement an algorithm they do not understand!*

Help is available!

Ask for help when you need it!

Asking for help is *not* an admission of failure. REALLY... it ISN'T.

Asking for help is a *normal* part of learning!!

Asking for help correlates with student success.

Asking for help is not a bother for the instructor/TAs. If you overwhelm us, we'll hire more TAs.

Expected Knowledge from CMPT 270

- Java
  - Object Oriented Programming
    - No static methods or variables (except very special situations)
    - Use of polymorphism.
    - Use of inheritance.
    - An understanding of Java's Generic Types.
  - Basic knowledge of Java API
    - Basic collection classes, (e.g. ArrayList, LinkedList)
    - Common interfaces, e.g. `Comparable`
    - Console and text file I/O
  - Use of Preconditions
- UML
  - Read UML diagrams.

Tools You Are Expected to Know

- IntelliJ IDEA (looks and feels same as PyCharm)
  - Use in development of projects.
  - Breakpoints and stepped execution for debugging (from 141 - same interface as PyCharm)
  - Free download of Community Edition from `jetbrains.com`.
  - Ultimate Edition also available to students for free with Jetbrains account. (Community edition is sufficient!)
  - Ultimate Edition available in Spinks labs.

Academic Honesty

- There is no group work in this course. All submissions are expected to be individual efforts.
- Where is the line?

  ✓ Talking about solutions with classmates at a high level.

  ✓ Helping each other with small compiler errors, debugging.

  ✗ Exchange written/electronic materials.

  ✗ Code while sitting next to your friend, peeking at each other's screens (I see this a **lot** in the labs!).

  ✗ No code photography.

- Computers make it **easy** to cheat.

Academic Honesty

- There is no group work in this course. All submissions are expected to be individual efforts.
- Where is the line?

  ✓ Talking about solutions with classmates at a high level.

  ✓ Helping each other with small compiler errors, debugging.

  ✗ Exchange written/electronic materials.

  ✗ Code while sitting next to your friend, peeking at each other's screens (I see this a **lot** in the labs!).

  ✗ No code photography.

- Computers make it **easy** to cheat.
- Computers make it **easy** to detect cheating.

Notes from the Undergraduate Chair

- If you got a failing grade in 270 last term, you must drop 280.
- Every year, a small number of students cannot graduate because their major average is below 62.5%. Often, low marks in 2nd year are the problem.
- If you have a mark in the 50s in a course, it can be retaken once, but only if done before taking a course that uses the course to be retaken as a prerequisite!
- E.g., A student with 55% in CMPT 270 should consider retaking it, but must retake it before taking CMPT 280 because 270 is a prerequisite for 280.
- A full slate of 200-level CMPT courses are being offered in summer 2018: 214, 215, 260, 270, 280

Notes from the Undergraduate Chair

- We strive to provide a professional, inclusive, and friendly environment in labs.
- Labs are often crowded and conversations are easily overheard.
- Students should conduct themselves in ways that are appropriate for a professional environment, i.e. a way that is suitable for the workplaces you are training to enter.
- Inappropriate behaviour may be subject to action under the University's standard of student conduct in non-academic matters.
- Episodes of inappropriate conduct in labs can and should be brought to the attention of and/or discussed with any CS prof, the undergrad chair, or the department head.

Course Delivery Style

- Short readings before lectures.
- Lots of class time devoted to problem solving - individual, small groups, and as a class.
- Same style as CMPT 141, 145.
- Come prepared! Lectures will assume you've done the readings, review of readings will be *minimal*.
- Assignments reinforce and elaborate on in-class concepts.
- Tutorials focus on practical application of in-class concepts, extra examples of in-class concepts and problem solving exercises, approaches to solving assignments, and **some** new material not presented in class of a practical nature.

Late Assignment Policy

- Late assignments are not accepted (see course syllabus).
- This includes "I tried to submit at the last minute and was cut off."
- **Do not wait until the last minute.**
- There are too many students in this class to accommodate the handing in of "barely missed" deadlines.
- That said, if your files get uploaded successfully before the cutoff, they are considered on time, even if Moodle labels them "late" (which can sometimes happen).

Next Class

- Reading for next class: Textbook, Chapter 1.
- Class textbook can be downloaded from the course website.

- Any questions before we adjourn?