CSCA48 Winter 2018
Week 1: ADT

Marzieh Ahmadzadeh, Nick Cheng

University of Toronto Scarborough

Welcome

- What is this course about?
  - More on ADT
  - Data structures such as linked lists, trees, etc.
  - Algorithm Analysis
  - Sorting and searching
  - Recursion

Means of Communications

- https://mathlab.utsc.utoronto.ca/courses/csca48/

- Link to lecture notes, assignments, exercises, Piazza, Markus, announcement, course syllabus etc.

Means of Communications

- Piazza
  - First point of contact
  - Link in course website
  - For any question about weekly assignments, technical problems or anything that peers or TAs can answer.
    - But not for major assignments
  - If you registered for the course but not in Piazza, drop me an email with your utorid, name and student number

- Be respectful
  - You may be anonymous for your peers but not for instructors.

Means of Communications

- Tutorials
  - To practice what you have learnt
  - To learn new materials
  - Smaller group of people
  - Pop up quizzes
  - Once a week
    - Attend on your tutorial only
  - No tutorial this week.
- Practicals
  - To practice even more
  - Drop-in sessions
  - Attend as many sessions as you'd like
  - No practical this week.
  - Time & location will be announced later.
- Office hours

Lectures

- Lecture slides will be posted on course webpage
- Attendance is strongly recommended
  - The usefulness of slides is similar to having some (but not all) of the ingredients of a Pizza without a dough
  - Important announcement will be made

References

- Online book: How to think like a computer scientist
- Textbook: Data structure and algorithm in Python by M. Goodrich
- Reading materials will be provided weekly on course webpage

Assessment

Assessment	Weight	Comment
Midterm	25%	Written Exam. Two tests.
Final	45%	Comprehensive written exam.
Major Assignments	15%	2 Assignments.
Weekly Assignments	15%	10 Assignments. Equal weight.

To pass the course, you must earn at least 50% of the total of midterm and final

Important Dates

		Uploaded Date	Deadline	
			Date	Time
Weekly Exercises	1	12-Jan	19-Jan	5:00 PM
	2	19-Jan	26-Jan	5:00 PM
	3	26-Jan	2-Feb	5:00 PM
	4	2-Feb	9-Feb	5:00 PM
	5	9-Feb	16-Feb	5:00 PM
	6	16-Feb	23-Feb	5:00 PM
	7	2-Mar	9-Mar	5:00 PM
	8	9-Mar	16-Mar	5:00 PM
	9	16-Mar	23-Mar	5:00 PM
	10	23-Mar	30-Mar	5:00 PM
Assignment 1		2-Feb	17-Feb	5:00 PM
Assignment 2		23-Feb	10-Mar	5:00 PM

Midterm 1	TBA
Midterm 2	TBA
Final	TBA

Collaboration

- Exercises are there to help you learn:
  - So I do encourage you to collaborate on exercises however everyone should submit their own version.
- Assignments are there to evaluate you:
  - So NO collaboration on assignments are accepted.
  - Plagiarism detectors will catch you!

Late submission

- No late submission for exercises is accepted.
- Late submission for assignments are accepted if you have legitimate reason such as medical emergency.
  - Email me asap before the deadline with illness verification form.

Missed midterm

- If a midterm is missed for legitimate reason
  1. Provide the document such as illness verification form
  2. If accepted, your final exam will replace your missed midterm.

A08, 108, A20 and A48

- If you passed 108 or A20, you need to fill the gaps.
  - I don’t know, which part(s) you will need a help on. So ask questions and let me know.
- If you passed A08, make sure you haven’t forgotten the key concepts.

Academic Integrity

- *Plagiarism - Using the words or ideas of another person without citing the source.*
- *Unauthorized Aids - Using unauthorized aids, which could be considered cheating on tests and exams.*
- *Unauthorized Assistance - Having someone else do the work for you.*
- *Forgery or falsification - Making a false statement, presenting a false document or signing someone else's name on a document required by the University.*
- *Personation - Having someone else write an exam for you or writing an exam for someone else.*
- *Self- Plagiarism - Submitting work for credit in a course when you have submitted it in another course*

AccessAbility

- Diverse learning styles
- disability/health consideration that may require accommodations
- Feel free to approach me and/or the Access*Ability* Services ASAP.
- Access*Ability* Services:
  - Room SW302
  - 416-287-7560
  - ability@utsc.utoronto.ca.
- The sooner you let us know your needs the quicker we can assist you in achieving your learning goals in this course

How to succeed in this course?

- Practice, practice, practice
- Do not miss anything in this course (i.e. Exercise, Assignment, Lecture, Tutorial, etc.)
- Do not just read the given codes, rewrite it yourself
- Ask questions!

Break

Why do we need an abstract class?

- To focus on operations rather than implementation
  - To hide away implementation details
- It doesn't make sense to instantiate some classes
  - abstract classes vs concrete classes
  - It makes it possible to declare a method without implementing it.
  - Therefore ADT is closely related to inheritance concept

Defining an abstract class in Python

- Abstract classes inherits from ABC (Abstract Base Classes) module in Python.
- At least one of the methods should be decorated with @abstractmethod
- An abstract class can contain both abstract and non-abstract method
- An abstract method must be overrriden in all of the subclasses.

Break

What is an ADT?

- An abstraction of a data structure.
- Includes:
  - Data
  - Operation
  - Exceptions (almost always)