NUS
National University of Singapore
School of Computing

CS2040 Data Structures and Algorithms

Course Admin

(AY2024/25 Semester 1)

Lecturers

- **Dr Chong Ket Fah**
  COM2-02-66
  chongket@comp.nus.edu.sg

- **Prof Roger Zimmermann**
  AS6-05-05
  rogerz@comp.nus.edu.sg

- **Dr Anandha Gopalan**
  COM2-03-21
  axgopala@comp.nus.edu.sg

Stuff you need

JDK (Java Development Kit) 21.0.4 (Need it to compile and run Java programs)

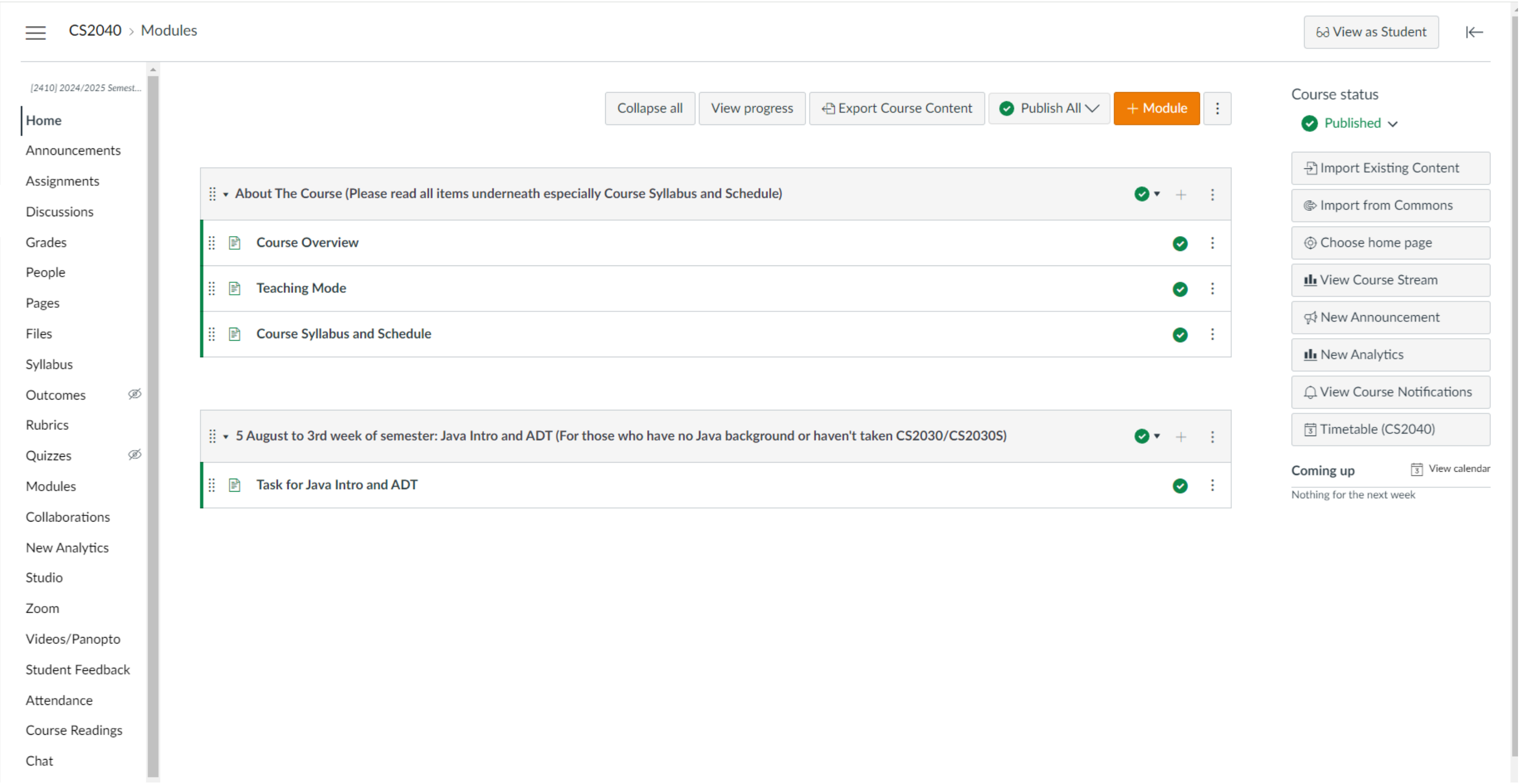
- https://www.oracle.com/java/technologies/downloads/#java21

Installation Guide for Windows/Linux/Mac OS

- https://docs.oracle.com/en/java/javase/21/install/overview-jdk-installation.html

CANVAS https://canvas.nus.edu.sg

- ❑ Announcements: Check regularly
- ❑ Files: For Lecture notes and tutorials
- ❑ Discussion Forum: Piazza forum (signup link is below)
  - https://piazza.com/nus.edu.sg/fall2024/cs2040



About The Course (Please read all items underneath especially Course Syllabus and Schedule)

- Course Overview
- Teaching Mode
- Course Syllabus and Schedule

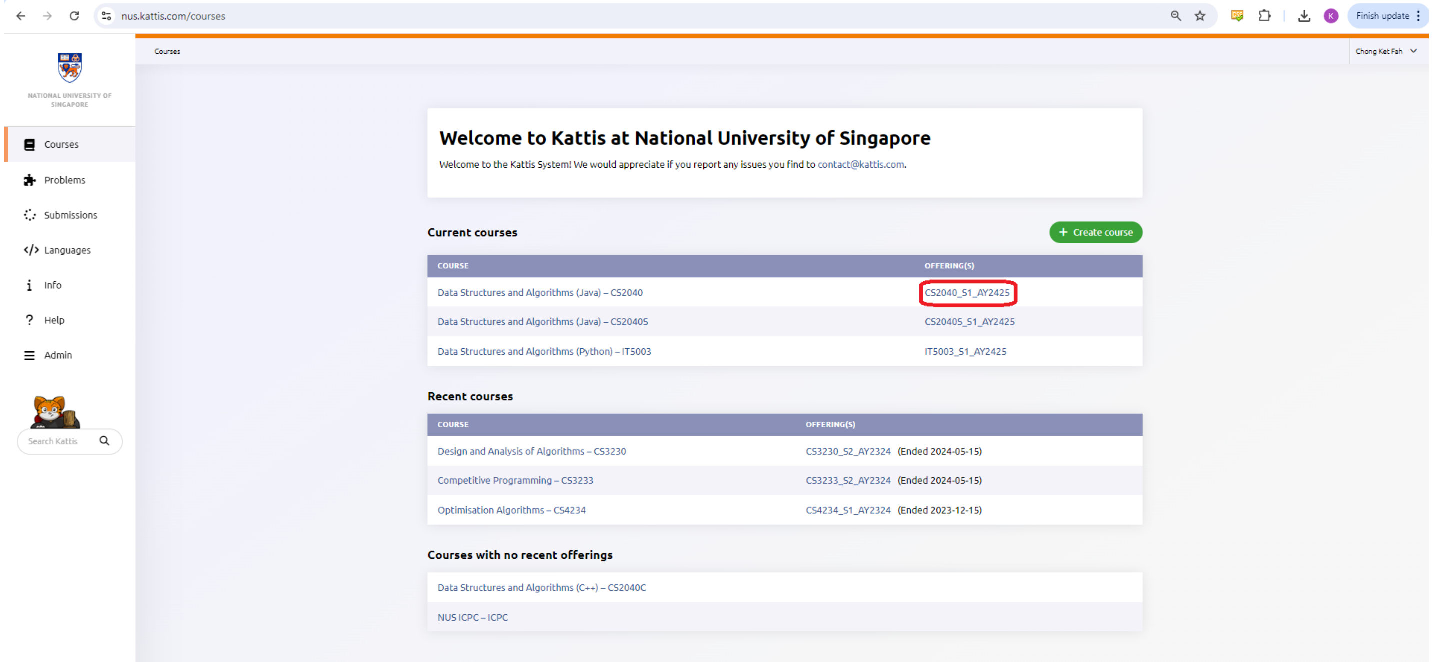
5 August to 3rd week of semester: Java Intro and ADT (For those who have no Java background or haven't taken CS2030/CS2030S)

- Task for Java Intro and ADT

Nothing for the next week

Kattis

https://nus.kattis.com/



Welcome to Kattis at National University of Singapore

Welcome to the Kattis System! We would appreciate if you report any issues you find to contact@kattis.com.

Current courses

COURSE	OFFERING(S)
Data Structures and Algorithms (Java) – CS2040	CS2040_S1_AY2425
Data Structures and Algorithms (Java) – CS2040S	CS2040S_S1_AY2425
Data Structures and Algorithms (Python) – IT5003	IT5003_S1_AY2425

Recent courses

COURSE	OFFERING(S)
Design and Analysis of Algorithms – CS3230	CS3230_S2_AY2324 (Ended 2024-05-15)
Competitive Programming – CS3233	CS3233_S2_AY2324 (Ended 2024-05-15)
Optimisation Algorithms – CS4234	CS4234_S1_AY2324 (Ended 2023-12-15)

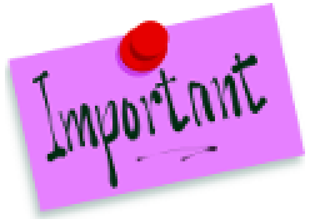
Courses with no recent offerings

Data Structures and Algorithms (C++) – CS2040C

NUS ICPC – ICPC

- Create an account here using your nusnet email (your eXXXXXXX@u.nus.edu email and not the one that uses your name as alias) and your username should be the same as your name that is shown in Canvas. If you already have an account or created one not following the requirement, please don’t create another, just use the existing account but inform your lab TA when lab starts about your username and email for the account.

- The registration key to register for the course will have been sent out to your Canvas inbox by the time of this lecture.



Other Important Links

Java API Specification Edition 8

(need to refer to it regularly in the course)

https://docs.oracle.com/javase/8/docs/api/

StackOverFlow

(find answers to most programming questions you have, but need to filter through a lot of information)

http://stackoverflow.com/

IDE for program development

http://www.sublimetext.com/download

Sublime Text is a simple general purpose IDE you may use for Java programming. If you have experience with other IDE’s you can use those too.

Reference Text

- CP4: Competitive Programming

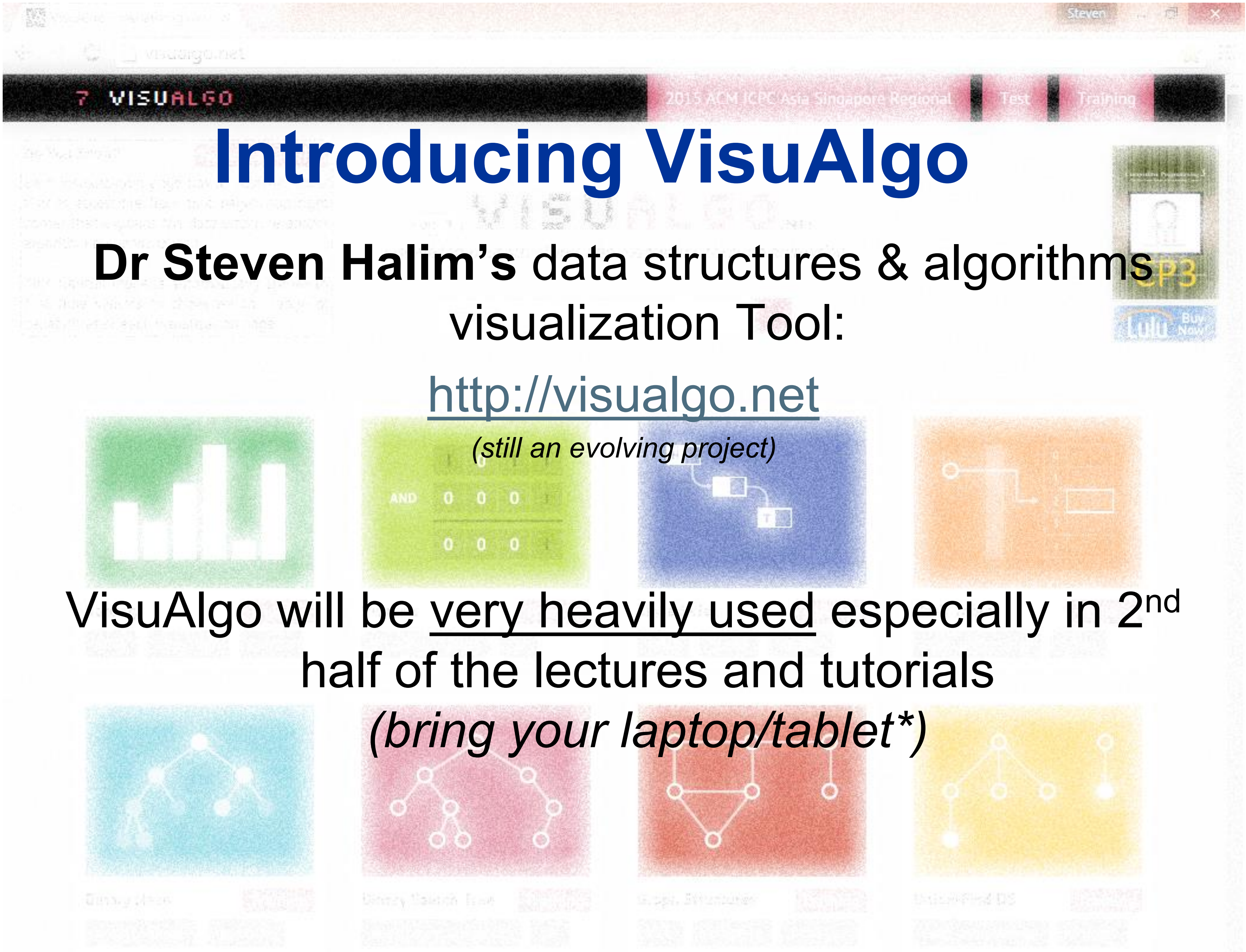
  Not compulsory (actually a CS3233 text book)

  Written by Dr Steven Halim and his brother Felix Halim

Competitive Programming 4
The Lower Bound of Programming Contests in the 2020s
Steven Halim, Felix Halim, Suhendry Effendy
Book 1
Chapter 1-4
Handbook for IOI and ICPC Contestants, and for Programming Interviews
Competitive Programming 4
The Lower Bound of Programming Contests in the 2020s
Steven Halim, Felix Halim, Suhendry Effendy
Book 2
Chapter 5-9
Handbook for ICPC and IOI Contestants, and for Computer Science enthusiast

- Available at Popular bookstores (e.g the one at Clementi)
- There are 2 books, book 1 and book 2, but only book 1 is most relevant for the course so buy only that one if you want the reference text (about $27)

Introducing VisuAlgo



Dr Steven Halim's data structures & algorithms visualization Tool:

http://visualgo.net

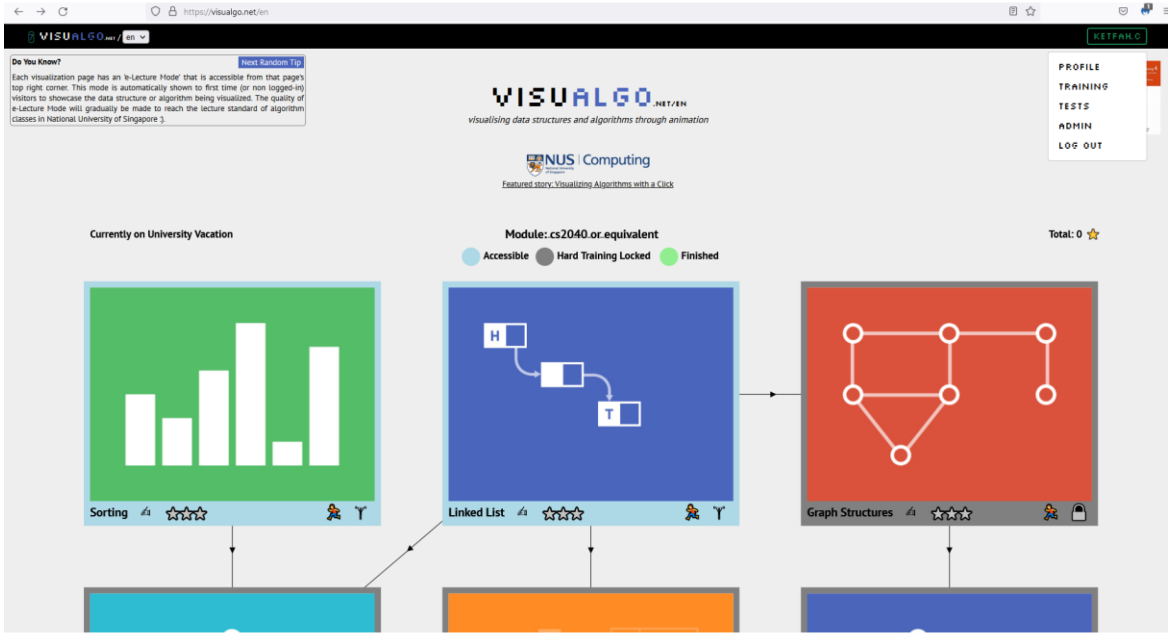
(still an evolving project)

VisuAlgo will be <u>very heavily used</u> especially in 2nd half of the lectures and tutorials

(bring your laptop/tablet)*

VisuAlgo Online Quiz Tool

1. An account will be created for you using your NUSNET email (eXXXXXXX@u.nus.edu email again) and you will get an email containing the password for logging into your account latest by end of week1 or start of week2. **Please don't create an account by yourself**
2. If you already have an account created using your NUSNET email, continue using it, no new account will be created again.
3. Once you have verified, you can log in and set your profile so that "Preferred Layout" is "cs2040 or equivalent"



https://visualgo.net/en

Do You Know?

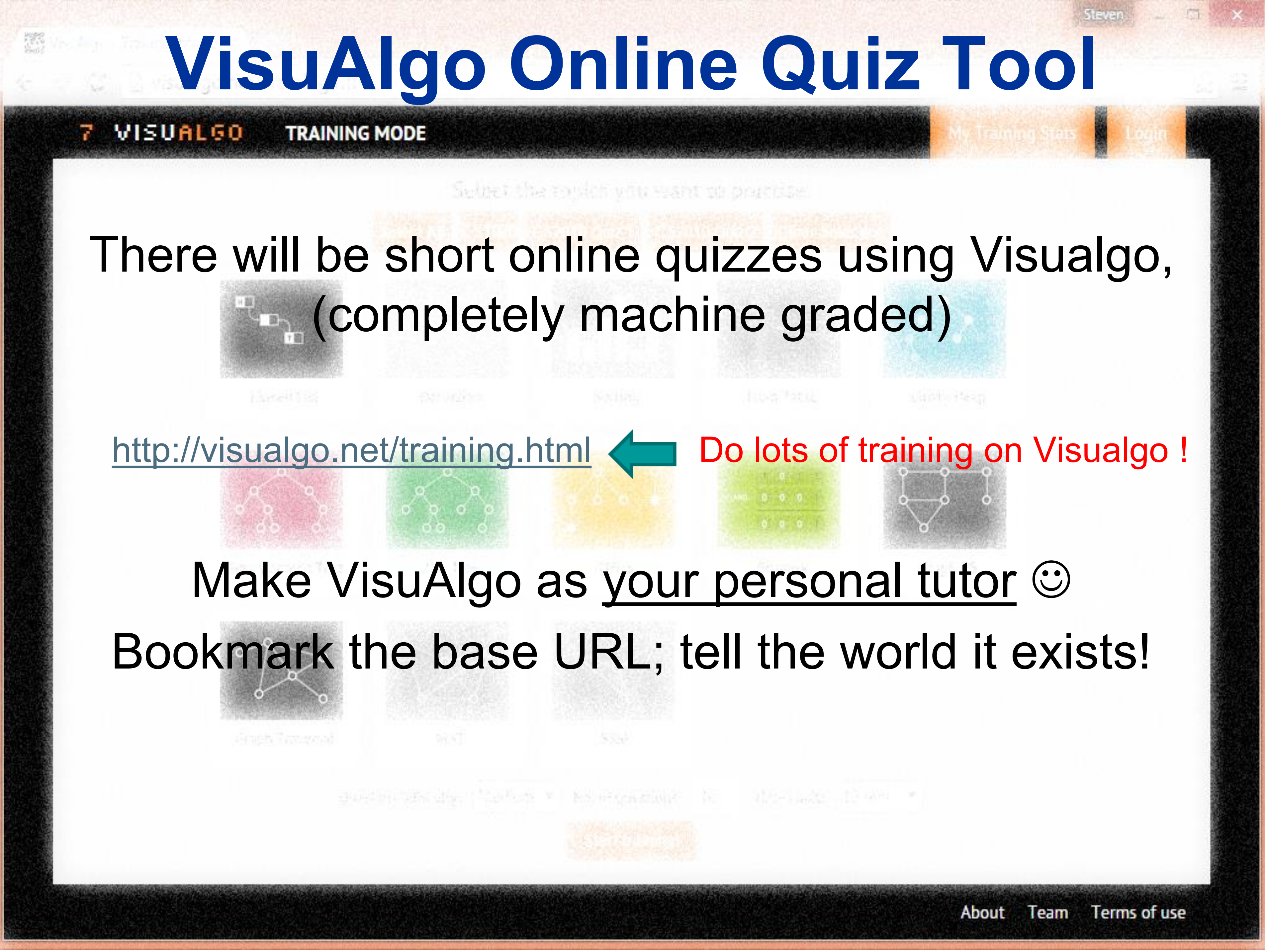
Each visualization page has an 'e-Lecture Mode' that is accessible from that page's top right corner. This mode is automatically shown to first time (or non logged-in) visitors to showcase the data structure or algorithm being visualized. The quality of e-Lecture Mode will gradually be made to reach the lecture standard of algorithm classes in National University of Singapore ;).

Currently on University Vacation

Module: cs2040 or equivalent

Total: 0 ☆

VisuAlgo Online Quiz Tool



There will be short online quizzes using Visualgo, (completely machine graded)

http://visualgo.net/training.html ← Do lots of training on Visualgo !

Make VisuAlgo as <u>your personal tutor</u> ☺

Bookmark the base URL; tell the world it exists!

Lectures, Tutorial, Lab Timings

- Lectures are online through Zoom, Tutorials and Labs are onsite.

- Lectures (a recording will be uploaded to Canvas)
  - Wednesday 10am-12noon
  - Thursday 9am-10am

- Tutorials (1 hr per session starting on 3rd week)
  - Thursday 8am to 6pm (check your tutorial group timing)

- Labs (2 hrs per session starting on 3rd week)
  - Friday 8am to 6pm (check your lab group timing)

- Course Syllabus and Schedule
  - Check on Canvas under Home → About the Course → Course Syllabus and Schedule

Zoom Consultation Sessions

- Starting from the 3rd week, there will be a weekly 1 hour long zoom consultation sessions for you to clarify any doubts or questions you have. The actual number of sessions and the timing will be announced at a later date.

Assessments: Overview

- 10 graded **1 day** lab assignments (1 assignment every lab starting from week 3) which will be released at 8am on Friday and end at 8am on Saturday the next day. (Solve 1 problem)
  - You can start doing when the problem is released
  - Everyone will have a lab on that day where the TA will talk about the problem, show Java classes useful to solve the problem and help you with the assignment (without directly giving you the answer)

- 4 graded take home lab assignments (Check schedule when they are released)
  - Will be released every Tuesday 10am of the week they are released
  - Deadline is usually due Tuesday 10am 2 weeks later
  - Solve 2 harder problems

- 2 online quiz (abt 30 mins)
  - Happen during lab on the 7th and 13th week (4th Oct and 15th Nov) respectively

Assessments: Overview

- 1 Midterm (5th Oct Saturday, 5:30pm-7:00pm tentatively)
- 1 Final (27rd Nov Wednesday, 5:00pm-7:00pm)
- midterm and final will be using Examplify.

Assessments: Overview

Activities	Weightages
Tutorial attendance/participation	3%
Lab attendance	2%
In-lab Assignments	15% (1.5%/problem)
Take Home Assignments	12% (1.5%/problem)
Online Quiz	8% (4% each)
Midterm	20%
Final Exam	40%

- Labs and Tutorials start on the **3rd week**.
- Online quiz, Midterm and Final exam are open-book (but not open internet)

Lab Assignment: Marking Scheme (1/2)

- Will use Kattis for autograding

- Calculation of grades for assignments (same day/take home) =

$$\left\lceil \frac{\#\ correct\ test\ cases}{total\ test\ cases} \times 1.5 \right\rceil \text{-(programming style violations)}$$

Lab Assignment: Marking Scheme (2/2)

- Programming style:
  1. Modularity
  2. Meaningful comments
     - Student particulars and program description
     - A description for each user-defined method
     - Appropriate pre- and post-conditions
     - Other comments to explain complex codes (where necessary)
  3. Meaningful/descriptive identifiers
  4. Proper indentation

- **0.5 mark** deducted if programming style is terrible (*make our eyes bleed*) on all of 4 main categories

- This means you should not have marks deducted unless your coding style is really terrible

Rules for Assignments (1)

1. The following actions are all taken as **PLAGIARISM**
   a. **Copy another person's code (fellow student or found on the internet)**.
   b. **Look at another person's code, understand it and then write your own code or modify from another person's code.** ← VERY IMPORTANT
   c. **Submit someone else's code just to check if it "passes the time limit"** *(all your submissions are logged so we can check …)*
   d. **Get someone other than your TA/lecturer to look at your code to debug it.**
   e. **Submitting to any alternate account you have created**. If you are caught "plagiarizing" yourself, I will still take it as plagiarism
   f. **Use AI tools to help in your assignment in ANY WAY** (not even to generate algorithm/pseudocode)**.**

Rules for Assignments (2)

2. You can discuss the **solution** to the assignments (1 day or take home) at the **algorithmic level** (i.e English description or pseudo-code) with your fellow student
   - **NO JAVA CODE OR ANY OTHER CODE MUST BE INVOLVED**
   - List down all your collaborators in your program file

3. The only code you can refer to/modify from is the code given to you with the lecture notes

4. You have to write/debug the Java code yourself! Labs are all about **individual** implementation of the algorithmic solution

5. **Offender caught plagiarizing will be referred to the Board of Discipline**

Summary and advice

- The labs focus more on your programming skills:
  - Ability to translate idea/algorithm into actual program
- Online quiz test your basic to intermediate understanding of the working of the algo/DS
- Midterm/Final exam focus more on your problem-solving skills:
  - Ability to understand and reason about the problem
  - Ability to apply your knowledge to formulate solution
- You need to spend time on:
  - Actually coding to improve your programming skill
  - Thinking deeply/exploring/do all your tutorials to hone your problem-solving skills as memorization does not help much
  - Asking questions! (Use the Piazza group/zoom consultation etc)

End of file