NUS
National University of Singapore
School of Computing

CS2040 Data Structures and Algorithms I

Lecture Note #0

Welcome and Course Admin

(AY2022/23 Semester 1)

Staff

- Lecturers
  - A/P Tan Sun Teck

    COM2-03-45

    tanst@comp.nus.edu.sg

    dcstanst@nus.edu.sg

    6516 2778

  - Prof Roger Zimmermann

    AS6-05-05

    rogerz@comp.nus.edu.sg

    dcsrz@nus.edu.sg

    65167949

Staff

- Teaching Assistant
  - Mr Ivan Chew Teck Meng

    COM2-02-20/21/43/44

    ictm@comp.nus.edu.sg

    ictm@nus.edu.sg

Part time TA

TAs: Mostly UG students who scored A/A+ for the module.

Time Table

- **Lecture**

  Wednesday 10am – 12noon

  Thursday 5pm – 6pm

- **Tutorial**
  - 18 groups all on Thursday
- **Lab Sessions**
  - **23 groups all on Friday**

Outline

1. Module Overview
2. Objectives
3. Resources
   - Luminus
   - Piazza
   - Reference book
4. Assessments
   - Tutorial
   - Laboratory
   - Quiz
   - PE
   - Midterm test
   - Final exam

1. Module Overview

- Some history:
  - CS1101 → CS1102 (before 2011)
  - CS1010 → CS1020 → CS2010
  - CS1010 → CS2040 (starting AY1819Sem1)
  - Emphasizes on *algorithms* and *data structures*

1. Module Overview

- Topics covered:
  - Problem Solving life Cycle
  - Basic algorithmic analysis
  - Abstract Data Type
  - Linear Data Structures
    - List, Stack and Queue
  - Recursion
  - Sorting methods
  - Hashing
  - Priority Queue – Heap
  - Binary Search Tree/AVL Tree
  - Graph

2. Objectives

- With this course, you should be able to:
  - Use object oriented modeling to formulate solution
  - Utilize appropriate simple data structures in problem solving
  - Understand recursion and data abstraction
  - Understand program efficiency through algorithm analysis

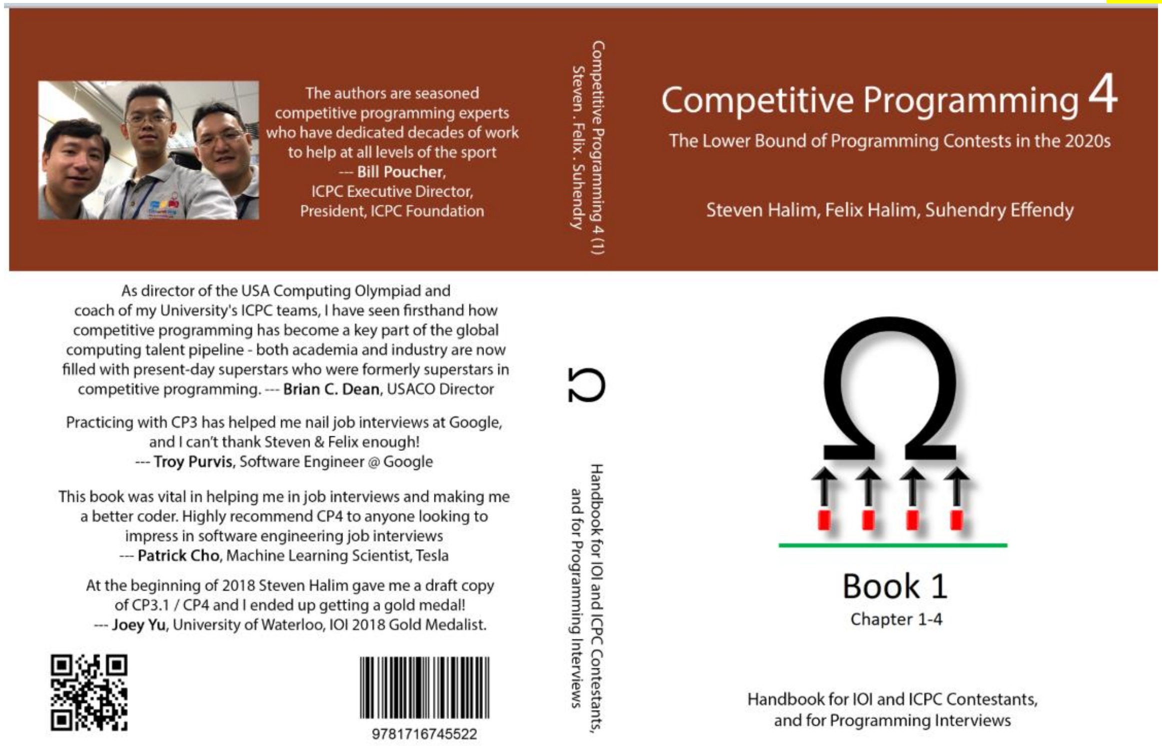
3. Resources: Luminus and Piazza

- Files: Lecturenotes. Tutorial and suggested solutions.
  Lab materials. Others.
- Lesson Plan: A guide. May need to change.
- Forums: In Piazza
  Use appropriate heading when you post. Check if someone has posted similar queries before you post.
- Announcements: The only communication channel.
  Check daily
- Gradebook: Make sure you verify when asked.

3. Resources: Reference Book

- CP4: Competitive Programming 4 – Book 1
  - Authors:
    Steven Halim, Felix Halim
    Suhendry Effendy

You can get the book from Forum Coop



The authors are seasoned competitive programming experts who have dedicated decades of work to help at all levels of the sport
--- **Bill Poucher**,
ICPC Executive Director,
President, ICPC Foundation

As director of the USA Computing Olympiad and coach of my University's ICPC teams, I have seen firsthand how competitive programming has become a key part of the global computing talent pipeline - both academia and industry are now filled with present-day superstars who were formerly superstars in competitive programming. --- **Brian C. Dean**, USACO Director

Practicing with CP3 has helped me nail job interviews at Google, and I can't thank Steven & Felix enough!
--- **Troy Purvis**, Software Engineer @ Google

This book was vital in helping me in job interviews and making me a better coder. Highly recommend CP4 to anyone looking to impress in software engineering job interviews
--- **Patrick Cho**, Machine Learning Scientist, Tesla

At the beginning of 2018 Steven Halim gave me a draft copy of CP3.1 / CP4 and I ended up getting a gold medal!
--- **Joey Yu**, University of Waterloo, IOI 2018 Gold Medalist.

9781716745522

Competitive Programming 4 (1)
Steven . Felix . Suhendry

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 1
Chapter 1-4

Handbook for IOI and ICPC Contestants,
and for Programming Interviews

3. Resources: Reference Book

- CP3: Competitive Programming 3
  - Authors:

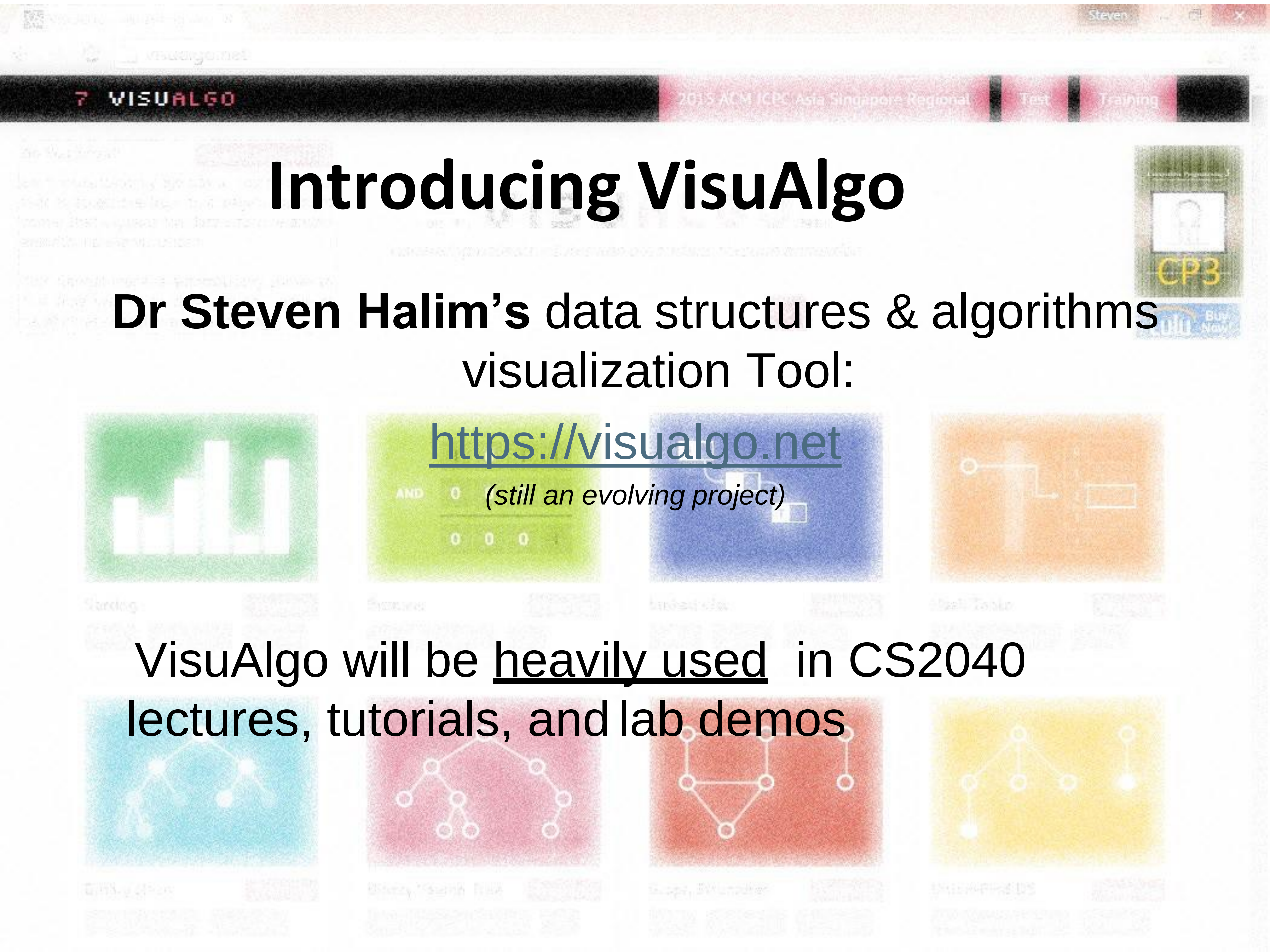
    Steven Halim

    Felix Halim
  - https://cpbook.net/#CP3details

https://docs.google.com/forms/d/e/1FAIpQLScYMs5yeEt9CKOuz7XhzhbCtJIsfcM-roT88irsGFQL6K4VxQ/viewform

Competitive Programming 3
The New Lower Bound of Programming Contests
Ω
Steven Halim
Felix Halim
HANDBOOK FOR ACM ICPC AND IOI CONTESTANTS
2013

Introducing VisuAlgo



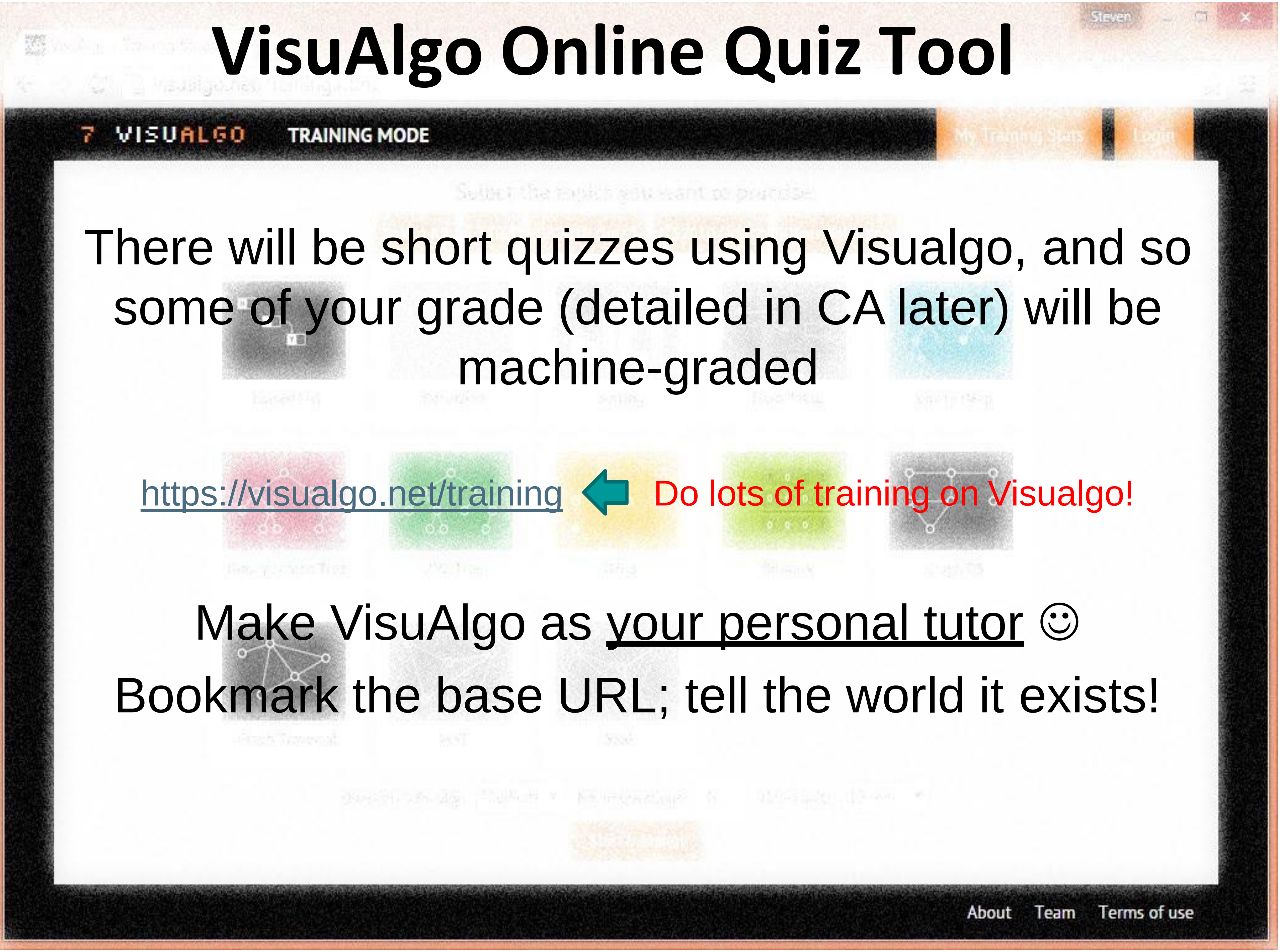
Dr Steven Halim's data structures & algorithms visualization Tool:

https://visualgo.net

(still an evolving project)

VisuAlgo will be heavily used in CS2040 lectures, tutorials, and lab demos

VisuAlgo Online Quiz Tool



There will be short quizzes using Visualgo, and so some of your grade (detailed in CA later) will be machine-graded

https://visualgo.net/training ⇦ Do lots of training on Visualgo!

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

Bookmark the base URL; tell the world it exists!

4. Assessment: Overview

- **CA 60%**
  - **Labs (open book) 12%**
    - More on this later
  - **Midterm test (closed book) 15%**
    - Date:1st Oct 2022， Saturday
    - Time: 10am to 12noon (1.5 hrs)
  - **Tutorial attendance/participation 6%**
  - **Visualgo Quiz 12%**
  - **Mock PE (open book) 3%**
    - 22nd Oct 2022, Saturday,
    - 2pm – 4pm (Two sessions)
  - **Practical Exam (open book) 12%**
    - 29th Oct 2022, Saturday,
    - 10am – 2.45pm (Two sessions)

4. Assessment: Overview

- **Final Exam (closed book) 40%**
  - Sat, 26 Nov 2022   13:00 - 15:00 (120 Minutes)

4.1 Tutorials

- Weekly, start from week 3
- You are expected to present solutions and participate in the discussion
- Suggested solutions will be released in the following week

4.2 Laboratory sessions

- Actual lab session starts from week 3
  - A special lab 0 has been released
    - Familiarize yourself with the system
    - Give away 1% for “free”
      - if you submit and pass all 3 exercises for lab 0
  - No deadline for lab 0

- Lab assignment:
  - **11 Take-home labs**
    - 1% per lab
- Visualgo Quiz: 2% x 6 (best 6 out of 9)

4.2 Laboratory:

- 3 Questions will be released 1 week before actual lab session
  - You are encouraged to attempt before going for the lab
- During the lab session, lab TA will:
  - Discuss possible approaches
  - Cover additional syntax (if any)
  - Demo implementation of one of the 3 questions
- You are expected to:
  - Submit a program for a graded task two days after the lab session
  - **Worth 1%**

4.2 Laboratory: Schedules

For the lab exercises in Week n.

- 3 problems will be given on Monday of Week n-1 on both Codecrunch and the lab folder in Luminus
- Download the task statement and test data from Luminus. Develop and test your programs in the Linus server or WSL and transfer them to your window directory
- Submit to CoreCrunch for automatic grading.
- Attend lab session in Week n for TA's advise.

Lab	Date	Type	Topics
0	Now (Week 0)	*Special*	
1	26 Aug (Week 3)	Ex #1	**Linear Data structure API**
2	2 Sept (Week 4)	Ex #2	**Linked list**
3	9 Sept (Week 5)	Ex #3 VQ #1	**Stack and Queue**
4	16 Sept (Week 6)	Ex #4 VQ #2	**Recursion**
5	30 Sept (Week 8)	Ex #5 VQ #3	**Sorting**
6	7 Oct (Week 9)	Ex #6 VQ #4	**Hashing**
7	14 Oct(Week 10)	Ex #7 VQ #5	**BST/AVL tree**
8	21 Oct(Week 11)	Ex #8 VQ #6	**Heap**
9	28 Oct(Week 12)	Ex #9 VQ #7	**PE Practices**
10	4 Nov (Week 13)	Ex #10 VQ #8	**Graph traversal**
11	11 Nov (Week 14)	Ex #11 VQ #9	**Graph**

4.2 Marking Scheme for graded task

- Programming style: 30%
  - Checked by Lab TAs
  - **Meaningful comments**: 10%
    - Purpose of methods and statements
    - Pre- and post-conditions
  - **Modularity**: 10%
  - **Meaningful identifiers**: 5%
  - **Indentation**: 5%
- You will be informed on which lab will be graded by TA.

Writing a class method for gcd

FractionV4.java

```java
public class FractionV4 {

  // Returns GCD of e and f
  // Pre-cond: e and f must be > 0
  public static int gcd(int e, int f) {
    int rem;
    while (f > 0) {
      rem = e%f;
      e = f;
      f = rem;
   }
   return e;
  }
  public static void main(String[] args) {

    // everything before computing gcd as in FractionV3

    int divisor = gcd(newNum,newDenom);
    newNum /= divisor;
    newDenom /= divisor;
    System.out.printf("New Fraction = "+newNum+"/"+newDenom);
  }
}
```

4.3 PE:

- Mock PE (22 Oct 2022, Sat 2pm to 4pm)
  - One easy question to be solved in one hour
  - Meant for preparing for the actual PE the week after.

- PE (29 Oct 2021, Sat 10am to 2.45pm)
  - 1 question (30%)
  - 1 question (70%)

4.3 PE:

- You will be given a different Plab account where all the necessary skeleton Java file, standard input and output files are given.
- Work on the given Java file. Do not change the name of the file and do not create new file.
- At the end of the session, you just need to log out of the Plab account.
- The Java file will be submitted to CodeCrunch for grading.

4.3 PE: Marking Scheme (cont)

- Correctness and efficiency: 70%
  - We will manually inspects your program
  - Partial credit will be awarded
- Penalties:
  - Non-compilable:
    - 50% off your final score (including both the style and correctness scores)
  - Empty program:
    - E.g. All codes are commented
    - Generally, commented code are ignored. So this could result in 0%

Academic Integrity

- Plagiarism is a serious offense.
  - Sharing your work with others also constitute plagiarism.
- Any form of cheating will be severely punished.
- If you need to do online exam, you must adhere to all the e-proctoring procedures. Any violation will be reported to university for disciplinary action.

Side effect of Academic offense

- You can NEVER be a teaching assistant.
- You may not be allowed to go for exchange.

Supplementary test and PE

- An absence will result in a ZERO mark unless a valid excuse with documentation is given
- A make-up PE and midterm test will be conducted
- Only those with proof will be qualified to attend
- The difficulty of the make-up may not be the same. Usually harder as you have more time to prepare.

Summary and advice

- The labs focus more on your programming skills:
  - Ability to translate idea into actual program
- Midterm and 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 skill
  - Thinking hard about the content of the lectures as memorization does not help

Learn to use UNIX

- Labs, Codecrunch and PE will be using UNIX based submission
- If you have time, you may find it useful to better learn the UNIX environment
- Useful tools
    - Standard UNIX tools
    - Text Editor (pico, vim, emacs)
    - File redirection (<,|,>)

Lab testing and I/O re-direction

- All the lab/PE exercise assume input from standard input (the keyboard) and output to standard output (the monitor).
- No interactive programming.
- To take input from a given input test case

```
$java progName < progName1.in
```

- To send output to a output file instead of the monitor

```
$ java progName > progName1.txt
```

- To check if your program pass the test

```
$ diff progName1.txt progName1.out
```

“Get to know you better” survey

Will provide the result when it is done

How the lectures will be conducted

- Flipped classroom
  - Pre-recorded lecture to be viewed before lecture time
  - Q&A during actual lecture time
- Cons
  - “The workload is supposed to be 3 hours per week, now you are asking us to spend more than 6 hours”
  - Some students might not keep pace
- Pros
  - No interruption during lecture
  - Total control in viewing lecture, maybe spend less time.
  - Keep lectures on schedule, impact on Labs and tutorials
  - The Q&A sessions allow you to prepare for midterm, PE and final exam in advance and not just the week before.

E-assessments