NUS
National University
of Singapore

School of Computing

CS1020 Data Structures and Algorithms I

Lecture Note #0

Welcome and Course Admin

(AY2016/17 Semester 2)

Staff

- Lecturers
  - A/P Tan Sun Teck
    COM2-03-02
    tanst@comp.nus.edu.sg
    dcstanst@nus.edu.sg
    6516 2778

Sectional 2

  - A/P Tan Tiow Seng
    AS6-04-10
    tants@comp.nus.edu.sg
    dcstants@nus.edu.sg
    6516 6764

Sectional 1

Main Teaching Assistant

- Ivan Chew Teck Meng
  - COM2-02-20

    ictm@u.nus.edu

    dcsctmi@nus.edu.sg

    65113103

Time Table

- **Lecture**
  - Tuesday 10am – 12nn
  - Sectional 1 @ LT15
  - Sectional 2 @ICube Auditorium
- **Tutorial**
  - **Venue: Seminar rooms in COM1/level 2**
  - Friday: 9-10, 10-11, 11-12, 12-1, 1-2, 2-3, 3-4, 4-5
  - Wednesday: 9-10, 10-11, 12-1, 1-2, 3-4
- **Lab Sessions**
  - **All labs are conducted on Thursday**
  - Sessions: 10-12, 12-2, 2-4, 4-6
  - Venues: Programming labs in COM1

Outline

1. Module Overview
2. Objectives
3. Resources
   - 3.1 Module website
   - 3.2 IVLE
   - 3.3 Textbook
4. Assessments
   - 4.1 Tutorial
   - 4.2 Laboratory

1. Module Overview

- This module is:
  - The second part of the 3-module introductory programming course
    - CS1010 → **CS1020** → CS2010
  - Emphasizes on *algorithms* and *linear data structures*
- Topics covered:
  - Object Oriented Programming (OOP) Model
    - Using Java programming language
  - Classic data structures
    - Lists, Stack and Queue
  - Recursion
  - Basic algorithmic analysis
  - Sorting methods
  - Hashing

2. Objectives

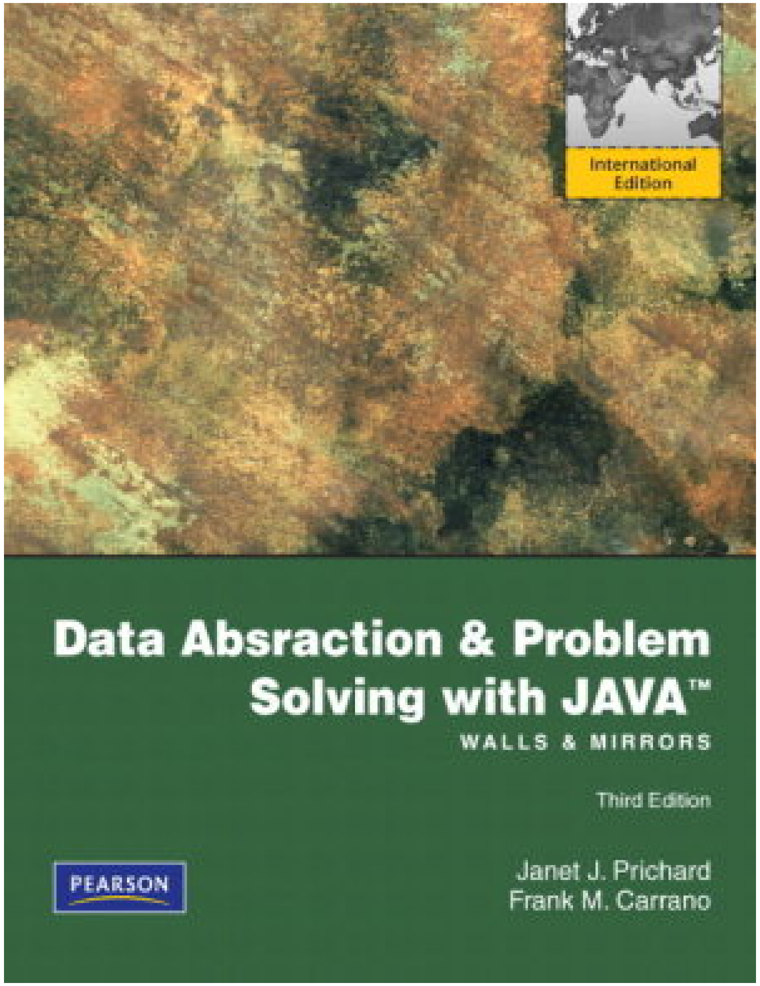
- With this course, you should be able to:
  - Use object oriented modeling to formulate solution
  - Utilize appropriate simple data structures in problem solving
  - Understand recursion and data abstraction
  - Understand program efficiency through algorithm analysis

3. Resources: IVLE

- IVLE
  - Workbins: Lecturenotes. Tutorial and suggested solutions. Lab materials. Others.
  - Lesson Plan: A guide. May need to change.
  - Forums: Use appropriate heading when you post. Check if someone has posted similar queries before you post.
  - Announcements: The only communication channel. Check daily
  - Anonymous feedback: Not allowed. We should be speaking our mind and be responsible for it.

3. Resources: Textbook

- Data Abstraction and Problem Solving with Java: Walls and Mirror
  - International edition, 3rd edition
  - Authors: Janet J. Prichard and Frank M. Carrano
  - Publisher: Pearson
  - ISBN: 9780273751205



4. Assessment: Overview

- **CA 60%**
  - **Sit-in Labs (open book) 18%**
    - More on this later
  - **Midterm test (closed book) 15%**
    - Date: 4th March 2017, Saturday
    - Time: 10am-12noon (tentative)
    - Venue: To Be Decided
  - **Tutorial attendance/participation 5%**
  - **Lab attendance 2%**
  - **Practical Exam (open book) 15%**
    - 1st Apr 2017, Saturday,
    - Session 1: 9.30pm to 12.00noon
    - Session 2: 12noon to 2.30pm
    - Labs in COM1

4. Assessment: Overview

- **Final Exam (closed book) 40%**
  - Saturday, 29-Apr-2017 (Afternoon)

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 will be released for week 2
    - Familiarize yourself with the system
    - Give away 1% for “free”
      - if you submit and pass all 3 exercise lab 0
- Two types of lab sessions:
  - **6 Take-home labs**
    - 1% per session, total 4%
  - **4 Sit-in labs**
    - 6% per session, total 18%
    - Best 3 labs out of 4

4.2 Laboratory: **Take Home Lab**

- There are a total of 6 labs
- Question will be released 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)
- At the end of the session, you are expected to:
  - Submit your work
  - **Worth 1% per session (4 best scores out of 6)**

4.2 Laboratory: **Sit-in Lab**

- There are a total of 4 sit-in labs
- A sit-in lab is like a mini practical exam to **test your programming skills**
  - Test on topic(s) covered in the previous take-home lab
- Each sit-in lab is:
  - 1 hour 40 minutes in duration and **worth 6%**
  - Open book, but limited to printed material only
- Your best 3 sit-in labs out of 4 will be chosen
  - **Total 18%**
- You will get a makeup only if:
  - You missed sit-in labs with valid reasons. Submit document proof to me.

4.2 Laboratory: **Schedules (Tentative)**

Lab	Date	Type	Topics
0	19th Jan (Week 2)	*Special*	
1	26th Jan (Week 3)	Take-home #1	IO / OOP
2	**2st Feb (Week 4)**	**Sit-in #1**	**IO / OOP**
3	9th Feb (Week 5)	Take-home #2	Java API
4	**16th Feb (Week 6)**	**Sit-in #2**	**Java API**
5	2nd Mar (Week 7)	Take-home #3	LinkedList
6	**8th Mar (Week 8)**	**Sit-in #3**	**LinkedList**
7	15th Mar (Week 9)	Take-home #4	Stack/Queue
8	**22th Mar (Week 10)**	**Sit-in #4**	**Stack/Queue**
9	29th Mar (Week 11)	Take-home #5	PE Practices
10	13th Apr (Week 13)	Take-home #6	Hashing

6th Apr Mark-up PE/Sit-in lab during the normal lab sessions

4.2 Sit-in Labs: Marking Scheme

- 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%
- Try to get the "free" 30% for every sit-in lab!
- It would be given if you scored 20% of the correctness marks.

4.2 Sit-in Labs: Marking Scheme (cont)

- Correctness and efficiency: 70%
  - Lab TA 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%

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 (<,|,>)

Supplementary tests 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.