CS1020 Data Structures and Algorithms I
Lecture Note #0

Course Admin
(AY2015/6 Semester 2)
Outline

Module Overview

Objectives

Staff

Resources

Schedules

Assessments
Module Overview

Object Oriented Programming (OOP) model
- Using Java

Classic data structures
- Lists, Stacks, Queues

Basic analysis of algorithm

Recursion
- More advanced than CS1010

Sorting algorithms
- More advanced than CS1010

Hashing

CS1010

OOP

CS1020

Linear Data Structures

CS2010

Algorithms
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 **data abstraction**
- Understand **recursion**
- Understand program efficiency through **analysis of algorithms**
Lecturers

- Module coordinator
  A/P Tan Sun Teck
  COM2-03-02
  tanst@comp.nus.edu.sg

- Sectional Group 2 @ ICube Auditorium

- Mr. Aaron Tan Tuck Choy
  COM1-03-12
  tantc@comp.nus.edu.sg

- Sectional Group 1 @ SR1
Module website

http://www.comp.nus.edu.sg/~cs1020

- Welcome to CS1020! (AY2015/6 Semester 2)
- Important links:
  - Java API Specification Edition 7
- This website is currently being updated for the coming semester. More information will be updated progressively. Thank you.

---

**IVLE**

- **Announcements:** Check daily
- **Forums:** Use appropriate heading when you post

https://ivle.nus.edu.sg

---

3. Please write **meaningful and concise header**. For example, a question or "Why is this so?".

4. Check old topics so that you do not introduce duplicate posts.

5. Duplicate, obsolete, or irrelevant postings will be removed or moved to the correct forum if necessary.

All the forums here are open to staff, seniors, and other students. A Tan Tuck Choy, Aaron
CodeCrunch
http://codecrunch.comp.nus.edu.sg
Textbook

- Data Abstraction and Problem Solving with Java: Walls and Mirror
  - International edition, 3rd ed
  - Authors: Janet J. Prichard and Frank M. Carrano
  - Publisher: Pearson
  - ISBN: 9780273751205
  - Available at NUS Co-op @ Forum

Textbooks for loan

- For needy students
- Please refer to IVLE forum for details
Textbook

Grab a friend to enjoy

ALL Pearson International & Global Edition Books

for every two copies of the same title purchased

Promotion period:
11th to 15th January 2016.
AY2015/6 Semester 2
Module Information - Schedules

Calendar

For complete academic calendar, see NUS calendar.
Recess week: 20 - 28 Feb.
Public holidays: 1 Jan (New Year Day), 8 - 9 Feb (CNY), 25 Mar (Good Friday).
CS1020 Exam: 3 May 2016, Tuesday, 1-3pm. (See Examination → Time-Table, Semester 2, AY2015/6)
Assessments: Overview

http://www.comp.nus.edu.sg/~cs1020/1_module_info/desc.html

- There will be 6 take-home labs (inclusive of lab #0), 4 graded sit-in labs and a practical exam (PE).
  - Take-home labs and sit-in labs are held in alternate weeks. Sit-in labs are open-book but electronic devices (eg: labtops and thumb-drives) are not allowed.

- Tutorials and labs start in week 3

- Mid-term test and final exam are closed-book (no cheat sheet allowed)

<table>
<thead>
<tr>
<th>Activities</th>
<th>Weightages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tutorial attendance/participation</td>
<td>5%</td>
</tr>
<tr>
<td>Lab attendance</td>
<td>2%</td>
</tr>
<tr>
<td>Take-home labs</td>
<td>5%</td>
</tr>
<tr>
<td>Sit-in labs</td>
<td>18%</td>
</tr>
<tr>
<td>Practical exam</td>
<td>15%</td>
</tr>
<tr>
<td>Mid-term Test</td>
<td>15%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>40%</td>
</tr>
</tbody>
</table>
Laboratory sessions

- See module website for updates
- Actual lab session starts from week 3
  - A special lab #0 (1%) will be released in week 1
    - Familiarize yourself with the UNIX system and vim

Two types of lab session:

- **Take-home labs**
  - 5 sessions (best 4 out of 5 sessions; total = 4%)
  - Total: 4% + 1% (lab #0) = 5%

- **Sit-in labs**
  - 4 sessions, 6% per session
  - Total: 18% (Best 3 out of 4 sessions)
Take-home Labs

- 6 take-home labs (including lab #0)
- Released on CodeCrunch
  - Each lab consists of 3 exercises
  - You should attempt them before attending the lab
  - Only one of them will be graded
- During the lab session, your lab TA will:
  - Discuss possible approaches
  - Cover additional syntax (if any) or other related exercises/topics
  - Lab attendance: 2%
- Each take-home lab (except lab #0) is worth 1%
  - Must be submitted to CodeCrunch BEFORE deadline
  - Must obtain an ‘A’ for the graded exercise.
**Sit-in Labs**

- There are **4 sit-in labs**
- A sit-in lab is like a mini practical exam to test your **programming skills**
  - Each sit-in lab is:
    - 1 hour 40 minutes in duration and **worth 6%**
    - Open book, but limited to **printed material** only
    - API will be available on the computer
- **Your best 3 sit-in labs out of 4 will be chosen**
  - **Total: 18%**
- **You will be allowed to take a makeup only if**
  - You missed 2 or more sit-in labs with **valid medical certificates or official excuses**
Sit-in Labs: Marking Scheme (1/2)

- **Correctness:** 70 marks
  - Input: 10% (Correctly read in all input and used them)
  - Output: 10% (Output format only, not about correct result)
  - Correctness: 50% (partial credit will be given)

- **Programming style:** 30%
  - Modularity: 10%
  - Meaningful comments: 10%
    - Particulars
    - A description for each user-defined method
    - Appropriate pre- and post-conditions
    - Other comments to explain complex codes
  - Meaningful/descriptive identifiers: 5%
  - Proper indentation: 5%

- Programming Style marks will be given only if you score at least 20 marks for correctness.
Penalties:

- 50% will be deducted if the submitted program has syntax error.
- Commented codes are ignored in general.
Practical Exam

- Date: 2\textsuperscript{nd} April
- Time: 10am to 3pm
- Venue: PL labs in COM1 basement
- Open-book, similar to sit-in labs
- Marking scheme is the same as sit-in labs
# Lab Schedules *(Tentative)*

Plan is tentative. Refer to module website for the most up-to-date plan.

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Type</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14\textsuperscript{th} Jan</td>
<td>Special</td>
<td>Intro Workshop</td>
</tr>
<tr>
<td>1</td>
<td>--</td>
<td>Take-home #0</td>
<td>Basic Java/IO</td>
</tr>
<tr>
<td>3</td>
<td>28\textsuperscript{th} Jan</td>
<td>Take-home #1</td>
<td>Basic Java/Array</td>
</tr>
<tr>
<td>4</td>
<td>4\textsuperscript{th} Feb</td>
<td>Sit-in #1</td>
<td>Basic Java/Array</td>
</tr>
<tr>
<td>5</td>
<td>11\textsuperscript{th} Feb</td>
<td>Take-home #2</td>
<td>OOP</td>
</tr>
<tr>
<td>6</td>
<td>18\textsuperscript{th} Feb</td>
<td>Sit-in #2</td>
<td>OOP</td>
</tr>
<tr>
<td>7</td>
<td>3\textsuperscript{rd} Mar</td>
<td>Take-home #3</td>
<td>Linked List</td>
</tr>
<tr>
<td>8</td>
<td>10\textsuperscript{th} Mar</td>
<td>Sit-in #3</td>
<td>Linked List</td>
</tr>
<tr>
<td>9</td>
<td>17\textsuperscript{th} Mar</td>
<td>Take-home #4</td>
<td>Stack/Queue</td>
</tr>
<tr>
<td>10</td>
<td>24\textsuperscript{th} Mar</td>
<td>Sit-in #4</td>
<td>Stack/Queue</td>
</tr>
<tr>
<td>11</td>
<td>31\textsuperscript{st} Mar</td>
<td>Take-home #5</td>
<td>PE Practice</td>
</tr>
</tbody>
</table>
Assumptions
Or what we assume you should have learned in CS1010/CS1010J/CS1010S/CS1101S

Topics in C / Java / Python / Javascript

Program development
- Writing pseudocodes
- Edit – compile – execute” cycle
- Step-wise refinement
- Hand-tracing codes
- Incremental coding
- Testing
- Debugging

Programming environment/tools
- Operating system: UNIX
- Editor: vim
- Debugger: (eg: gdb)

Problem solving
- Class exercises
- Practice exercises
- Lab assignments
Summary and advice (1/2)

- 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 deep and exploring as **memorization does not help**
  - **Asking questions!** (Use the IVLE forums.)
We provide you

- Practice exercises on CodeCrunch
- Self-assessments (quizzes) on IVLE
- Help sessions (on request)

But, ultimately…

YOU must be prepared and willing to put in a lot of efforts!
Introductory Workshop

- Those of you who have taken CS1010/CS1010J are familiar with UNIX system and vim.

- For those who did not take the above and hence are unfamiliar with UNIX and vim, please attend an Intro Workshop on 14th January, Thursday, at PL2 (COM1 basement)
  - Session 1: 10am – 11:40am
  - Session 2: 12nn – 1:40pm
  - Session 3: 2pm – 3:40pm

- Please refer to IVLE forum “Intro Workshop” and sign up there
End of file