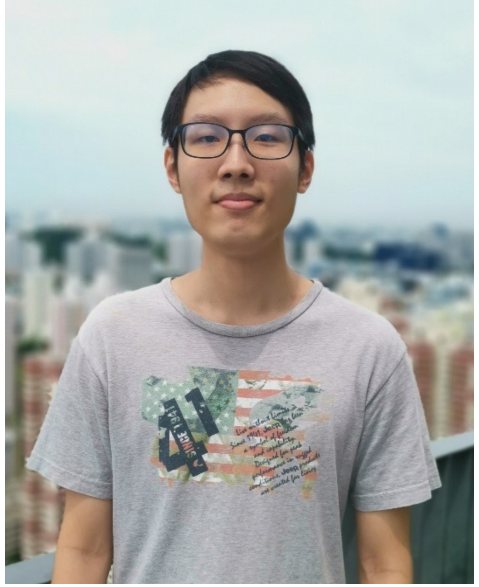
CS2040S Lab 1
Java Introduction

Lab TAs



Jon Chua

Faculty of Computing

jonchua@u.nus.edu

Huang Po Wei

Faculty of Computing

e0602114@u.nus.edu

Lab Introduction

- 2-hour session every Thursday
- Intended to provide hands-on experience with programming
- Involves one-day assignments, and take-home assignments
    - One-day assignments (solving 1 problem) should ideally be completed within the lab itself. The actual duration of the deadline is **10am on Thursday to 10am on Friday**, but the assignment is “*doable*” within the duration of the lab.
    - Take-home assignments (solving 2 problems, excluding optional challenge problems) should be completed before the deadline (2 weeks per take-home assignment)

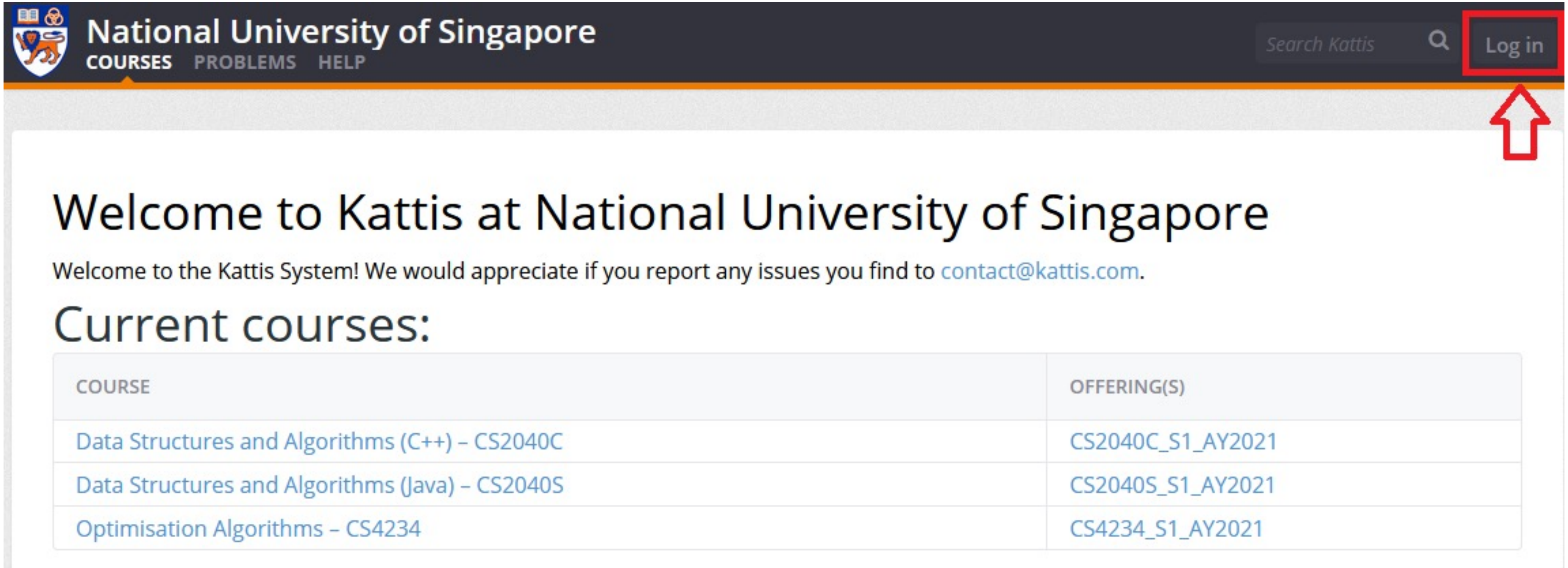
Lab Structure

- 1st part of labs (max. 30 min): short lesson
  - Cover answers for the previous session's assignments (one-day, and take-home, where applicable)
  - Cover relevant Java API for the session's given topic
  - Brief discussion of the one-day assignment
- 2nd part of labs: solving the one-day assignment (graded)
  - The first 20-30 minutes of this part will be used for students to plan how to solve the assignment, and express it in terms of pseudocode
  - Allowed to discuss at algorithm level with other students, but no discussing/sharing of code
  - **Type / write out the pseudo-code for your solution which must be shown to the TAs**
  - Can continue working on the assignment after the lab if necessary

Kattis Introduction

- Online platform used for submitting and grading assignments
- Found at https://nus.kattis.com/

Kattis Introduction – Signing Up



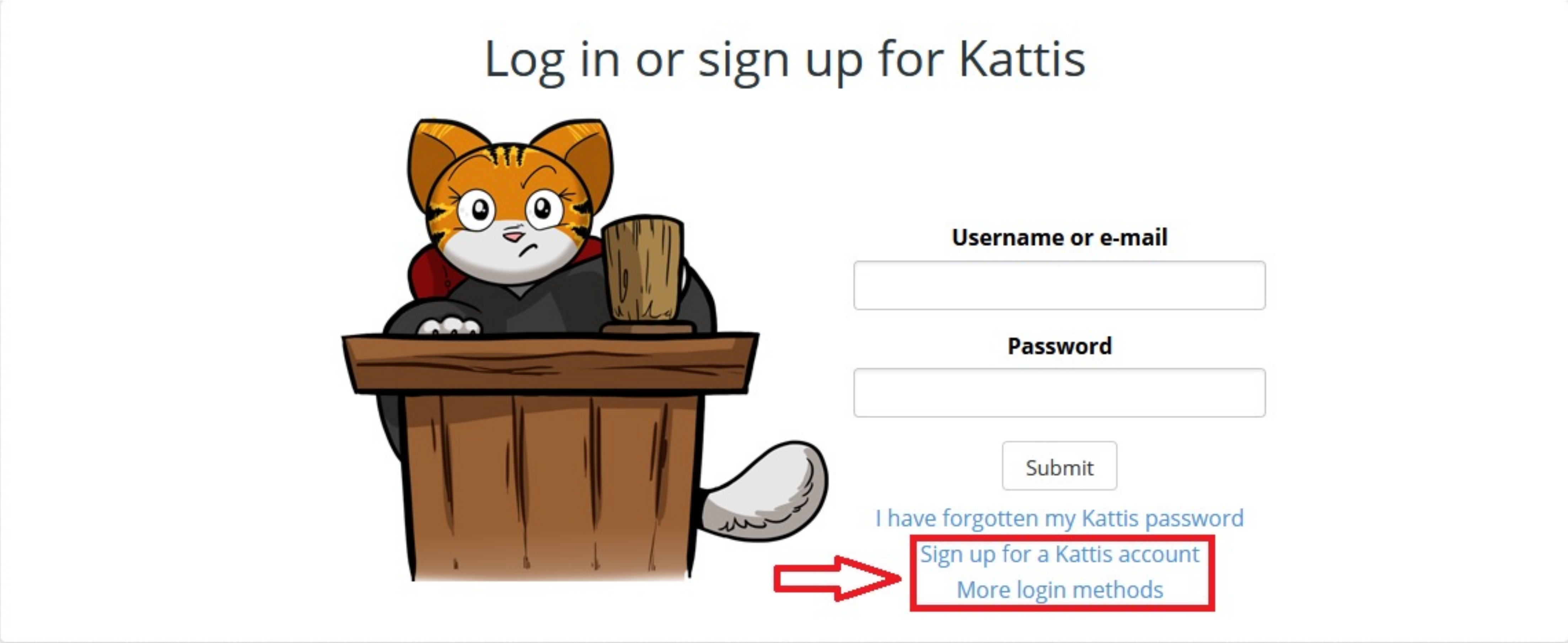
Welcome to Kattis at National University of Singapore

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 (C++) – CS2040C	CS2040C_S1_AY2021
Data Structures and Algorithms (Java) – CS2040S	CS2040S_S1_AY2021
Optimisation Algorithms – CS4234	CS4234_S1_AY2021

Kattis Introduction – Signing Up



Log in or sign up for Kattis

Username or e-mail

Password

I have forgotten my Kattis password

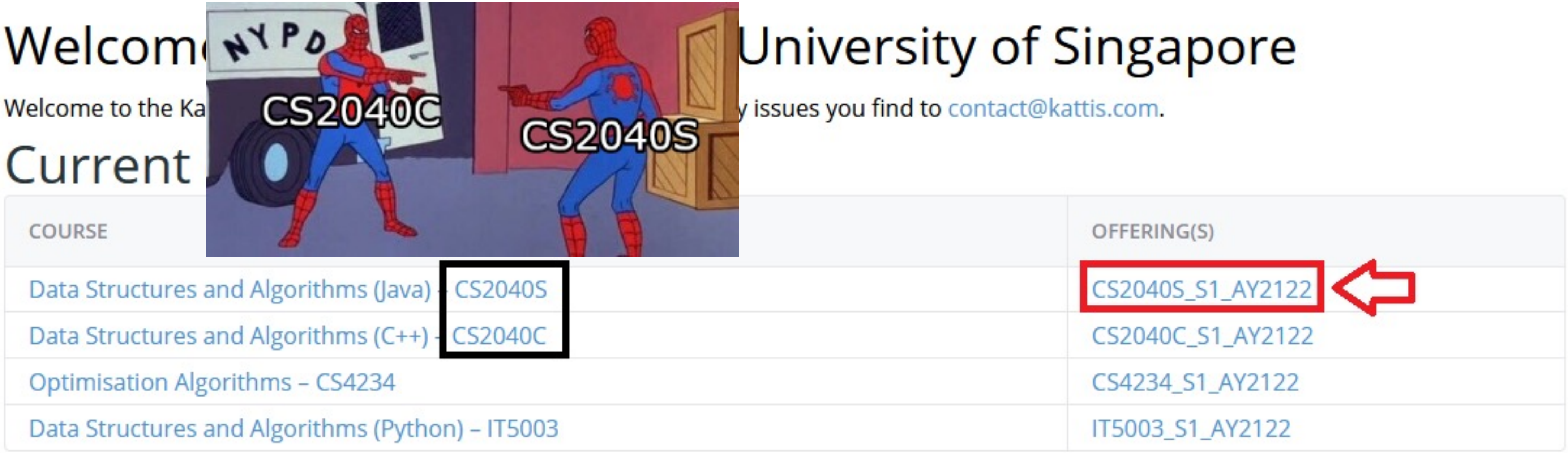
Sign up for a Kattis account

More login methods

Kattis Introduction – Signing Up

- Can sign up using email, or through linking to Facebook/Google/LinkedIn
- For name, please use your name as on LumiNUS, but only capitalize the first letter
- Eg. if your name on LumiNUS is “CHEN JIN XIONG, RYAN” (not a real student, as far as I can tell) then use “Chen Jin Xiong, Ryan” (ie, do not change the ordering)
- There is currently no way to edit an account name once it has been created, so be careful

Kattis Introduction – Registering for CS2040S



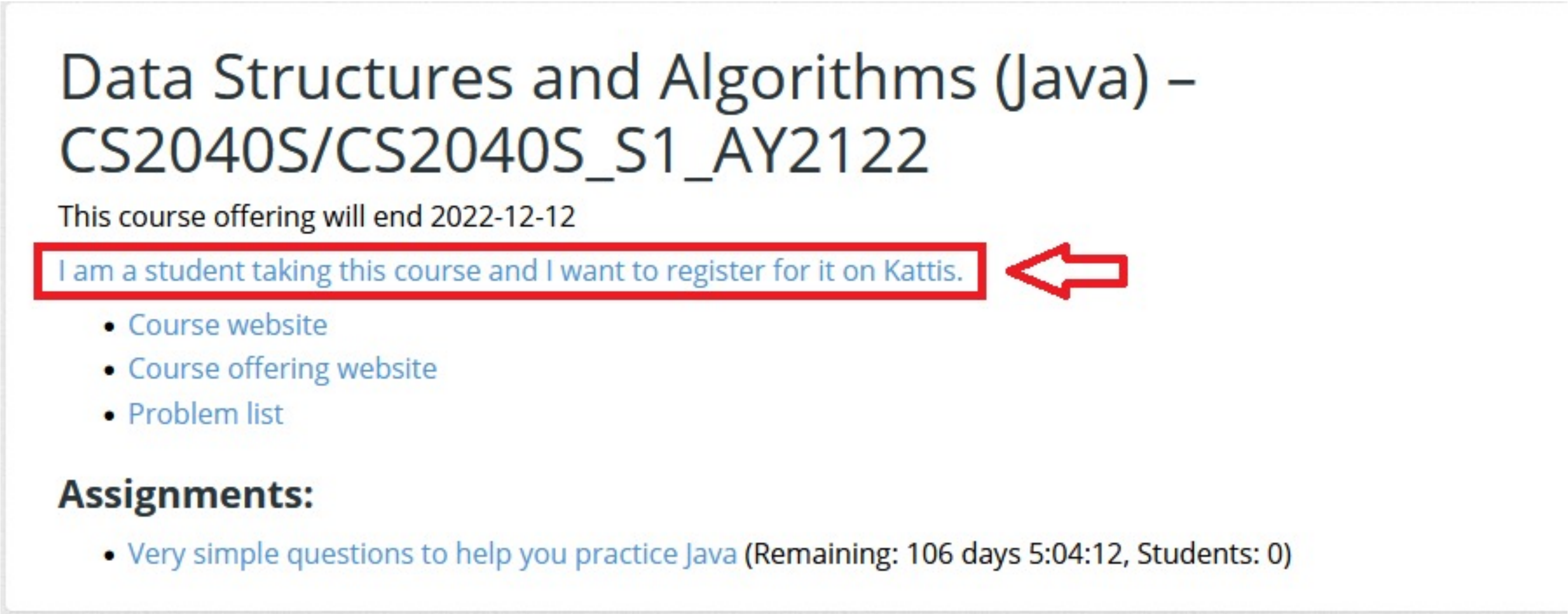
Welcom University of Singapore

Welcome to the Ka y issues you find to contact@kattis.com.

Current

COURSE	OFFERING(S)
Data Structures and Algorithms (Java) - CS2040S	CS2040S_S1_AY2122
Data Structures and Algorithms (C++) - CS2040C	CS2040C_S1_AY2122
Optimisation Algorithms – CS4234	CS4234_S1_AY2122
Data Structures and Algorithms (Python) – IT5003	IT5003_S1_AY2122

Kattis Introduction – Registering for CS2040S



Data Structures and Algorithms (Java) – CS2040S/CS2040S_S1_AY2122

This course offering will end 2022-12-12

I am a student taking this course and I want to register for it on Kattis.

- Course website
- Course offering website
- Problem list

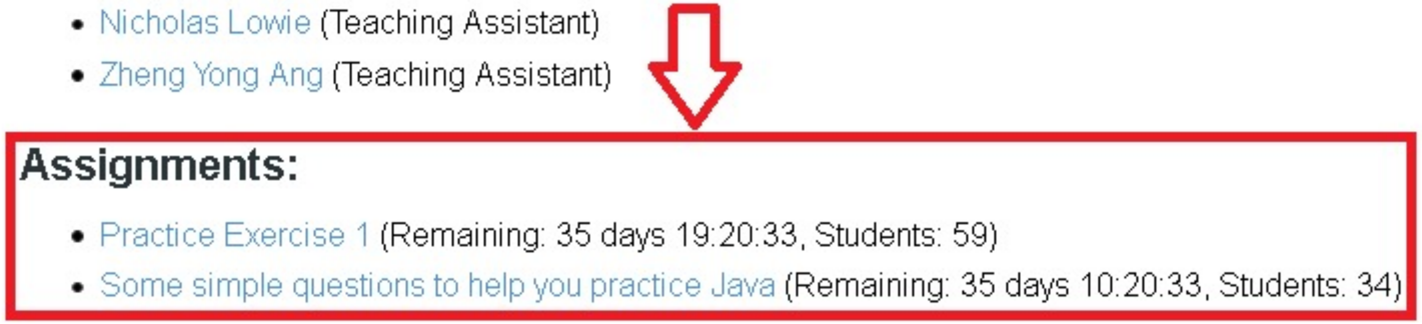
Assignments:

- Very simple questions to help you practice Java (Remaining: 106 days 5:04:12, Students: 0)

Registration Key = ckf_cs2040s_sem1_21_22

Kattis Introduction – Assignments

All problem sets will appear here (bottom of the course page)

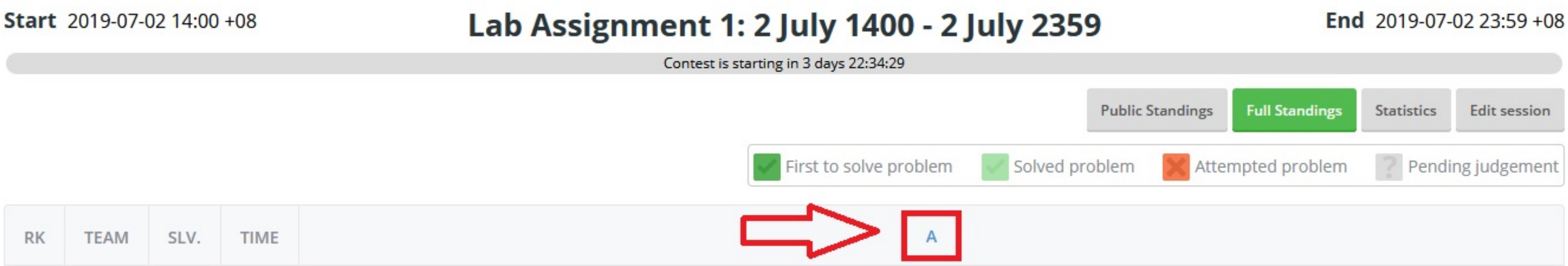


- Nicholas Lowie (Teaching Assistant)
- Zheng Yong Ang (Teaching Assistant)

Assignments:

- Practice Exercise 1 (Remaining: 35 days 19:20:33, Students: 59)
- Some simple questions to help you practice Java (Remaining: 35 days 10:20:33, Students: 34)

Kattis Introduction – Assignments



Start 2019-07-02 14:00 +08

Lab Assignment 1: 2 July 1400 - 2 July 2359

End 2019-07-02 23:59 +08

Contest is starting in 3 days 22:34:29

✓ First to solve problem ✓ Solved problem ✗ Attempted problem ? Pending judgement

RK	TEAM	SLV.	TIME	A

Click on the letter to access the problem (only visible when session opens)

Lab 1 – Runtime Analysis

- In CS2040, sometimes having a program/algorithm give the correct answer may not be sufficient; it may be required that your program/algorithm should run quickly as well
- Kattis provides the maximum running time for a program in the problem description on the right, under “CPU Time limit”
- Generally, machines on Kattis can run close to 100 million operations a second
- Try calculating whether your program can pass the time limit based off its time complexity
- Eg for a problem where N <= 1000, and the time limit is 1 sec, $O(N^3)$ = 1 billion = probably not okay (> 100 million), whereas $O(N^2)$ = 1 million = okay

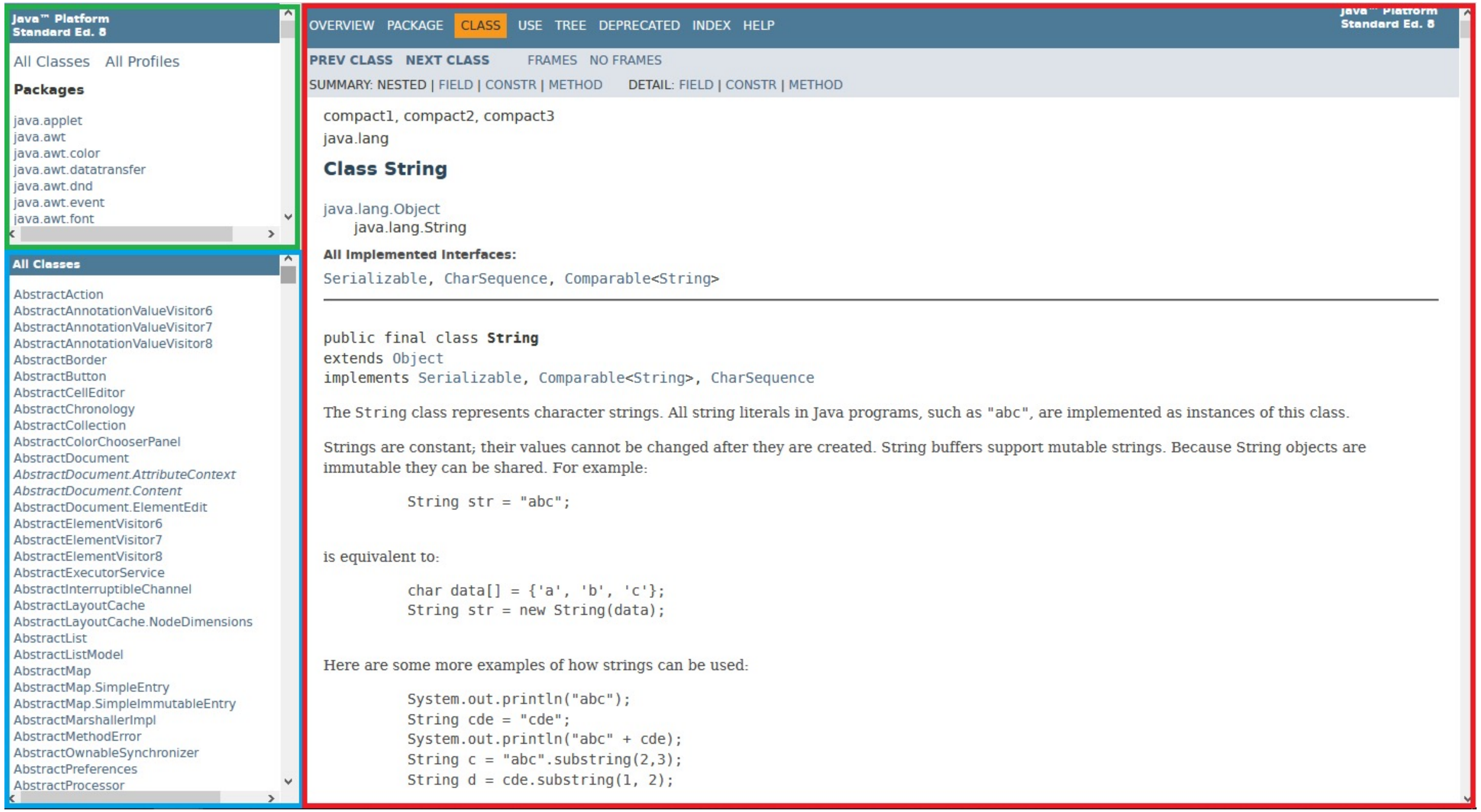
Lab 1 – Runtime Analysis

- Also note that the methods provided by Java API may not run in O(1) time, and as such your program may run slower than intended if this is not taken into consideration
- To fully understand the methods provided by the API, it is recommended to read up the documentation
- The following slides are based off the Java 8 API documentation: https://docs.oracle.com/javase/8/docs/api/
  - Documentation of the later versions of Java (up to Java 10) follows the same idea, but uses a different layout by default. To view them in the same manner, click on "FRAMES" at the top of the page
  - Documentation from Java 11 onwards does not seem to support frames, requiring manual navigation of the packages (or just search for the class name via Google, or the search bar at the upper right of the Java API page)

Lab 1 – Reading Java API

Package window: can click on them to view only classes from a single package

Class window: click on one of them to view the full API of that class



Packages

java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font

All Classes

AbstractAction
AbstractAnnotationValueVisitor6
AbstractAnnotationValueVisitor7
AbstractAnnotationValueVisitor8
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractChronology
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeContext
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractElementVisitor6
AbstractElementVisitor7
AbstractElementVisitor8
AbstractExecutorService
AbstractInterruptibleChannel
AbstractLayoutCache
AbstractLayoutCache.NodeDimensions
AbstractList
AbstractListModel
AbstractMap
AbstractMap.SimpleEntry
AbstractMap.SimpleImmutableEntry
AbstractMarshallerImpl
AbstractMethodError
AbstractOwnableSynchronizer
AbstractPreferences
AbstractProcessor

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP
Java™ Platform
Standard Ed. 8
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

compact1, compact2, compact3
java.lang

Class String

java.lang.Object
    java.lang.String

All Implemented Interfaces:

Serializable, CharSequence, Comparable<String>

```
public final class String
extends Object
implements Serializable, Comparable<String>, CharSequence
```

The String class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

```
String str = "abc";
```

is equivalent to:

```
char data[] = {'a', 'b', 'c'};
String str = new String(data);
```

Here are some more examples of how strings can be used:

```
System.out.println("abc");
String cde = "cde";
System.out.println("abc" + cde);
String c = "abc".substring(2,3);
String d = cde.substring(1, 2);
```

Main window: shows all the details of the selected class

Lab 1 – Reading Java API

- For the package window, clicking on a package will switch the class window to show only the classes in that package
    - Frequently used packages are:
        - java.util
            - Almost everything that's covered later in the module
        - java.lang
            - Strings and related classes
            - Wrapper classes
        - java.io
            - Buffered input/output (covered in a later lab)
    - For Java 9 onwards, these are found under the java.base package in the documentation (but retain the same package names otherwise)

Lab 1 – Reading Java API

- For the main window, the contents tend to be sorted as follows:
  - 1. Brief description of class
  - 2. Fields, constructors and methods (short version, sorted alphabetically)
    - Click on an entry here to go to the relevant entry in section 3
  - 3. Fields, constructors and methods (long version, may not be sorted)
- The time complexity of a method will sometimes be stated in section 1 or 3

Lab 1 – Useful API

- Scanner class – used for reading input
- Found in the java.util package; need to use the following line to import:
  - `import java.util.*;`
- Declare a new Scanner object with the following line (in main method):
  - `Scanner sc = new Scanner(System.in);`
- Read in input using the methods found in Scanner:
  - `int testCases = sc.nextInt();`

Lab 1 – Scanner

Method name	Description	Time
.nextInt()	Reads the next token in the input as an integer	O(n)
.nextDouble()	Reads the next token in the input as a double	O(n)
.next()	Reads the next token in the input as a String	O(n)
.nextLine()	Reads until it reaches the end of the line	O(n)

n refers to the length of the input that is read

These slides covering API will cover the most frequently used (but not all) methods of a class;
for a full list, refer to the Java API documentation

Lab 1 – System.out (not really a class)

Method name	Description	Time
System.out.print(String str)	Prints *str*	O(n)
System.out.println(String str)	Prints *str*, followed by a newline character (‘\n’)	O(n)
System.out.printf(String format, Object… args)	Emulates the printf function from C	O(n)

n is the length of the output

Technical note: System.out is an instance of the PrintStream class, so you may refer to the API documentation on PrintStream (in java.io) to explore more methods

Lab 1 – Useful API

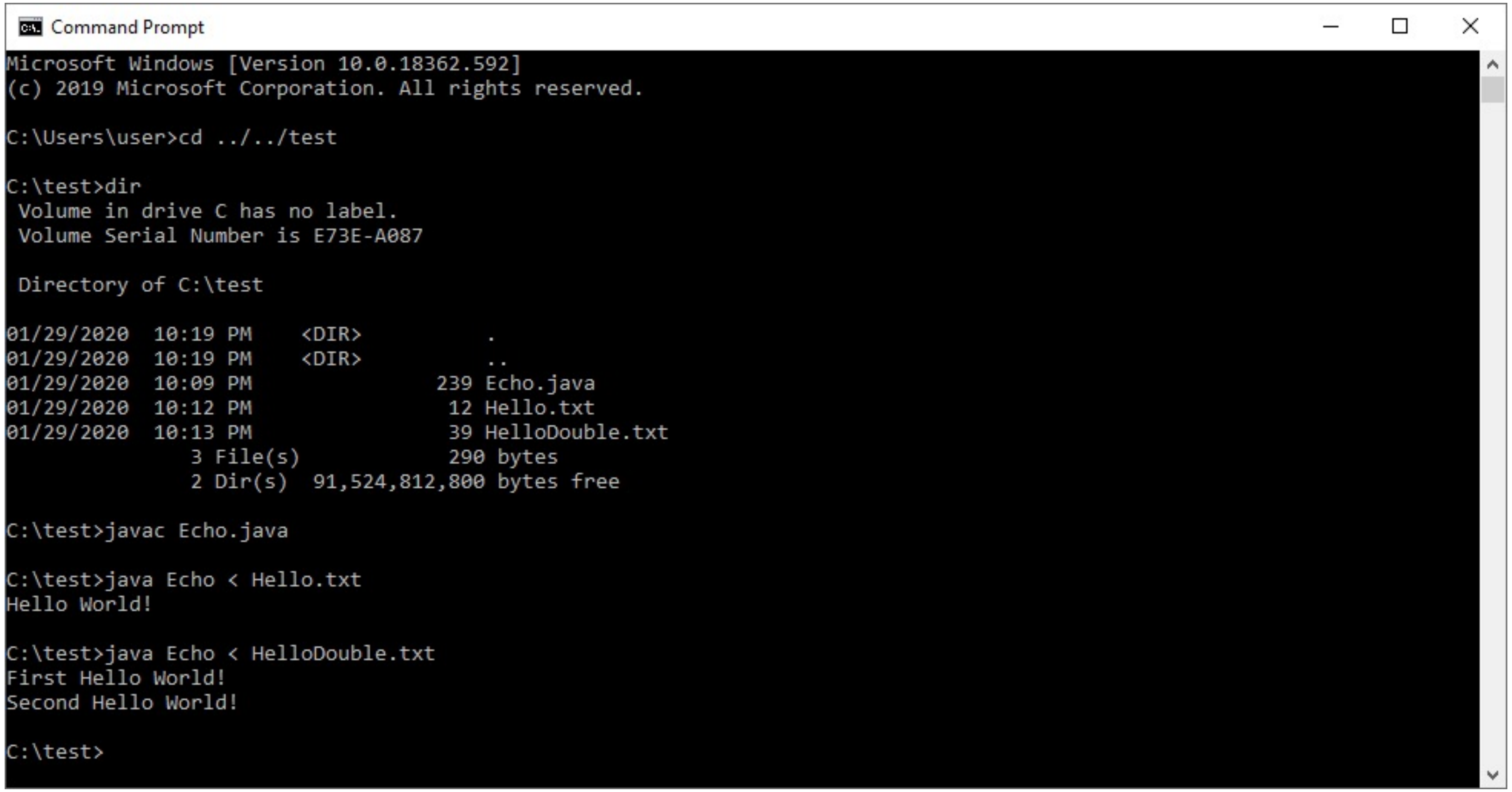
- String class – contains several methods that could be of use
- Found in the java.lang package; is imported by default
- Can be constructed in various ways:
  - `String str1 = new String("apple");`
  - `String str2 = "apple";`
  - `char[] arr = {'a', 'p', 'p', 'l', 'e'};`
  - `String str3 = new String(arr);`

Lab 1 – String

Method name	Description	Time
.split(String delim)	Splits a String into an array of Strings, based off the characters found in *delim*	O(length of string)
.charAt(index i)	Returns the character at index *i* (0-based)	O(1)
.equals(String other)	Checks if this string has the same value as the value of String *other* Note: not the same as using ==	O(length of shorter string)
.concat(String other)	Returns a new string which is this string + *other* Note: does not modify the original String (Strings are immutable in Java)	O(length of resulting string)
.length()	Returns the length of the string	O(1)
.substring(int start, int end)	Returns a new string, which contains the content of the original string from index *start* (inclusive) to index *end* (exclusive) (indices are 0-based)	O(length of resulting string)

Lab 1 – Command Line/Terminal Usage

- It is helpful to know how to compile and execute Java programs via command line (example below)



```
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\user>cd ../../test

C:\test>dir
 Volume in drive C has no label.
 Volume Serial Number is E73E-A087

 Directory of C:\test

01/29/2020  10:19 PM    <DIR>          .
01/29/2020  10:19 PM    <DIR>          ..
01/29/2020  10:09 PM               239 Echo.java
01/29/2020  10:12 PM                12 Hello.txt
01/29/2020  10:13 PM                39 HelloDouble.txt
               3 File(s)            290 bytes
               2 Dir(s)  91,524,812,800 bytes free

C:\test>javac Echo.java

C:\test>java Echo < Hello.txt
Hello World!

C:\test>java Echo < HelloDouble.txt
First Hello World!
Second Hello World!

C:\test>
```

Lab 1 – Command Line/Terminal Usage

- cd ../../test
  - “cd” is the command used to move between folders. “..” means to go up one folder
- dir
  - Used to list the contents of a folder. Use “*ls*” (lowercase L, lowercase S) on Unix-like systems or Powershell
- javac Echo.java
  - “javac” is the command use to compile Java programs. The full file name (inclusive of “.java”) is required, and seeing no output is a **good** thing.
  - Each class in the Java file is compiled to its own “.class” file

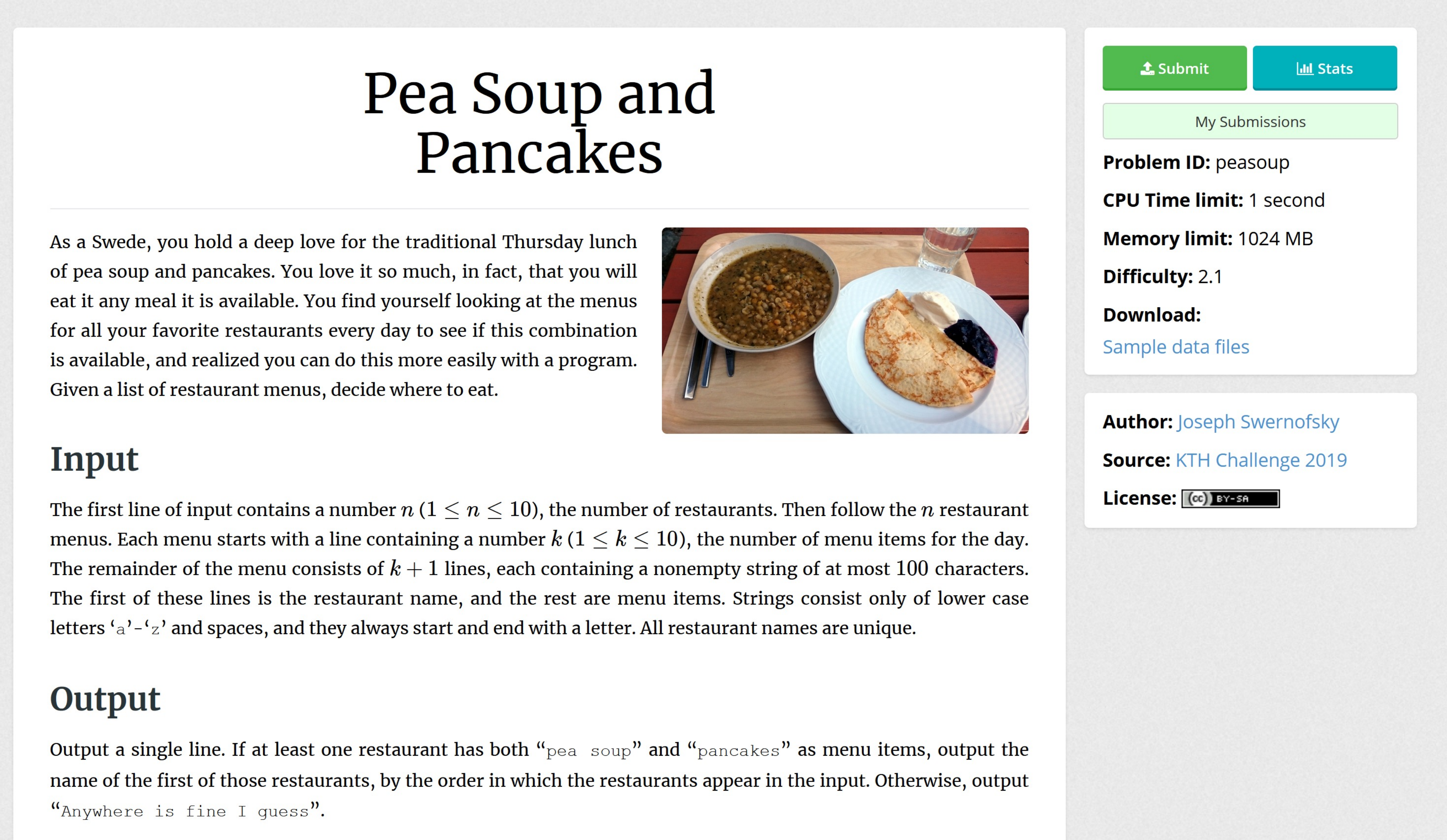
Lab 1 – Command Line/Terminal Usage

- java Echo < Hello.txt
- java Echo < HelloDouble.txt
  - “java” is the command used to run compiled Java programs. No filename extension (.java, .class) should be provided
  - “<“ followed by the name of a file is used to provide input from a file (known as input redirection).
    Instead of having to type out your input manually; simply save your input in a file and use “<“ to avoid retyping input each time you test your program
- java Echo > Bye.txt
  - “>” followed by the name of a file is used to send output to the file (known as output redirection)
  - Can be used together with input redirection, e.g. *java Echo < Hello.txt > Bye.txt*

One-Day Assignment 0 – Pea Soup

- First one-day assignment; is ungraded
- Found at https://nus.kattis.com/problems/peasoup
- Future assignments will be found on the main Kattis course page at https://nus.kattis.com/courses/CS2040S/CS2040S_S1_AY2122.
- Writing pseudocode for this assignment is not necessary, but will be required from the next one-day assignment onwards

Kattis Problems



Pea Soup and Pancakes

As a Swede, you hold a deep love for the traditional Thursday lunch of pea soup and pancakes. You love it so much, in fact, that you will eat it any meal it is available. You find yourself looking at the menus for all your favorite restaurants every day to see if this combination is available, and realized you can do this more easily with a program. Given a list of restaurant menus, decide where to eat.

Input

The first line of input contains a number n $(1 \leq n \leq 10)$, the number of restaurants. Then follow the n restaurant menus. Each menu starts with a line containing a number k $(1 \leq k \leq 10)$, the number of menu items for the day. The remainder of the menu consists of $k + 1$ lines, each containing a nonempty string of at most 100 characters. The first of these lines is the restaurant name, and the rest are menu items. Strings consist only of lower case letters 'a'-'z' and spaces, and they always start and end with a letter. All restaurant names are unique.

Output

Output a single line. If at least one restaurant has both "pea soup" and "pancakes" as menu items, output the name of the first of those restaurants, by the order in which the restaurants appear in the input. Otherwise, output "Anywhere is fine I guess".

Problem ID: peasoup

CPU Time limit: 1 second

Memory limit: 1024 MB

Difficulty: 2.1

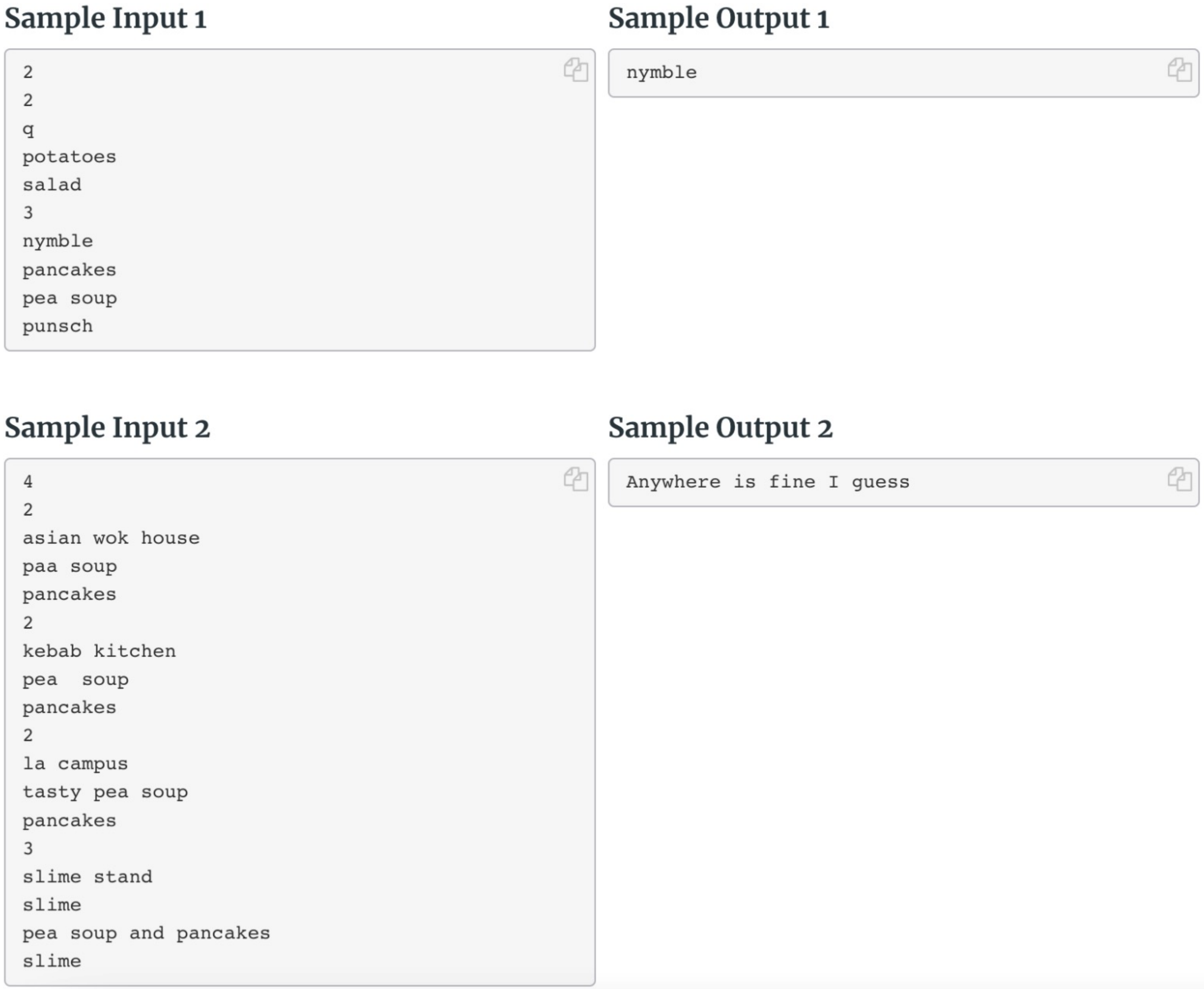
Download:
Sample data files

Author: Joseph Swernofsky

Source: KTH Challenge 2019

License: (cc) BY-SA

Kattis Problems



Sample Input 1

```
2
2
q
potatoes
salad
3
nymble
pancakes
pea soup
punsch
```

Sample Output 1

```
nymble
```

Sample Input 2

```
4
2
asian wok house
paa soup
pancakes
2
kebab kitchen
pea  soup
pancakes
2
la campus
tasty pea soup
pancakes
3
slime stand
slime
pea soup and pancakes
slime
```

Sample Output 2

```
Anywhere is fine I guess
```

One-Day Assignment 0 – Pea Soup

- Given a list of restaurants, and their menus, determine the first restaurant appearing in the list that offers both “pea soup” and “pancakes”
- Should be the exact strings “pea soup” and “pancakes”; even if a menu item contains the substring “pea soup” or “pancakes”, ignore it (see Sample Input 2)

Assignment Guidelines

- **Important:** Include your name and student number in comments at the top of your code

Rehash of “IMPORTANT Rules for CS2040S Assignments”:

- If you discuss the problem with any other student(s), include their name(s) as collaborators in a comment at the top of your code
- No usage of anyone else’s code (outside of code provided in lecture materials)
  - Directly using (eg. copy-paste) is not allowed
  - Using code as a reference is not allowed either

How to write pseudocode?

- https://www.unf.edu/~broggio/cop2221/2221pseu.htm

```
Set total to zero
Set grade counter to one
While grade counter is less than or equal to ten:
    Input the next grade
    Add the grade into the total
Set the class average to the total divided by ten
Print the class average.
```

Tips for CS2040

- More concerned about correctness and runtime complexity
- Not about modelling the problem statement using OOP (e.g. CS2030)
  - `Restaurant, Menu` class etc not required!
- It is (usually) ok to do everything in the `main()` class