CS1010S Programming Methodology

Lecture 1

Introduction to CS1010S & Python

15 Aug 2018

Welcome


Happy National Day

Selamat Hari Raya Haji

Makeup Lecture:

Mon 20 Aug, 6:30-8:30pm, UT-AUD2

3 things

WHY

should you take CS1010S

WHAT

to expect in CS1010S

HOW
to learn Python
(In 15 mins)

But first… The Cast

Wai Kay

Wai Home 1

About

Overview

Work and Education

Works at National University of Singapore

waikay@nus.edu.sg

April 2

Chat (34)

Dr Zhou Lifeng

Adi Yoga

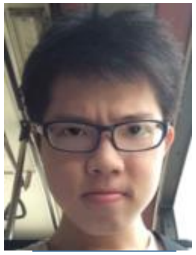
Oana Barbu

Prof Ben Leong



Tutors











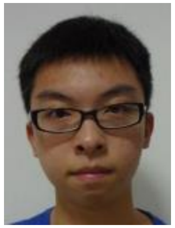


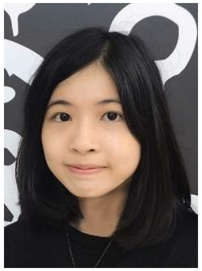




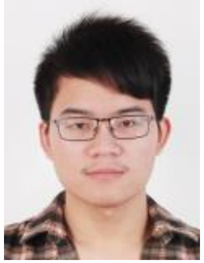


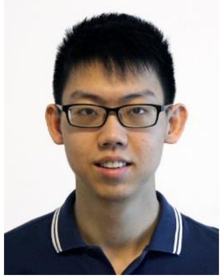
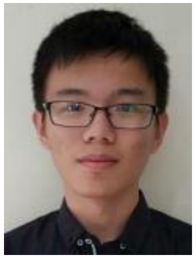









WHY

should you take CS1010S

WHY

are you taking

CS1010S

WHY are you taking CS1010S?

1. Because I love programming.
2. Because I want to learn programming.
3. Because I am interested in computing.
4. Bo bian, core requirement.

Some Statistics

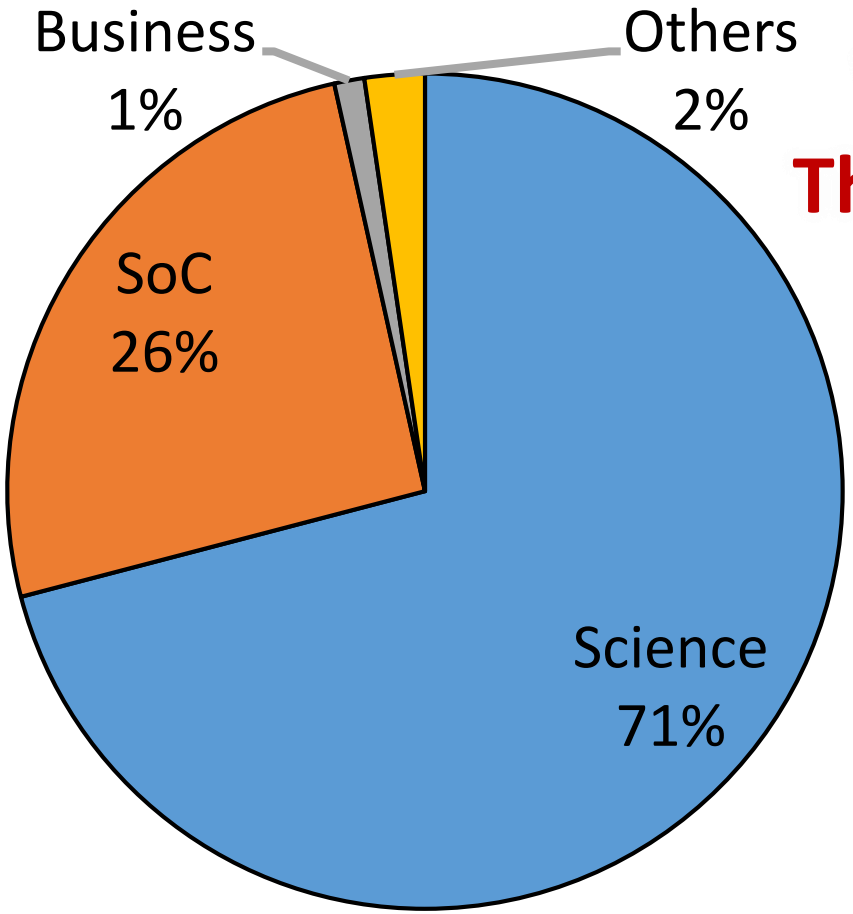
CORS

Round	Quota	Bidders	Winning Bid
Round 1A (Returning)	90/100	8/12	1/1
Round 1C (Returning)	81/88	90/89	156/1
Round 1C (New)	108/112	177/159	338/496
Round 2A	0/0	-	-
Round 2B	0/0	-	-
Round 3	0/0	-	-

Class full

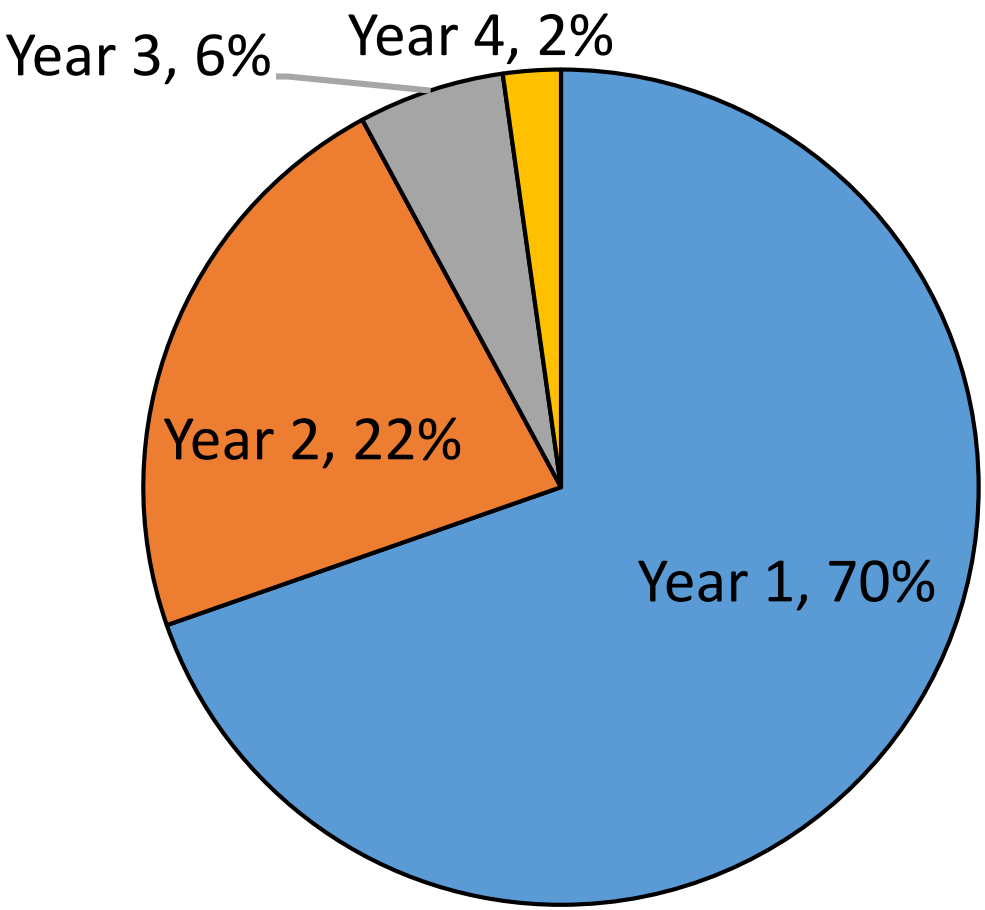
Quota taken up by preallocation

% by Faculty



Warning!
There will be Math!

By Seniority



WHY you should take CS1010S

1. Because computers are useful

1. Because computers are useful but they are dumb

They will precisely execute every instruction given, even bad ones

Personal Computer

What will you use to

- Write a report? 
- Do accounting? 
- Statistical analysis? 
- Find prime numbers?
- Solve a puzzle?

When is Cheryl's Birthday?

Cheryl's birthday is one of 10 possible dates.

May 15	May 16	May 19
June 17	June 18	
July 14	July 16	
August 14	August 15	August 17

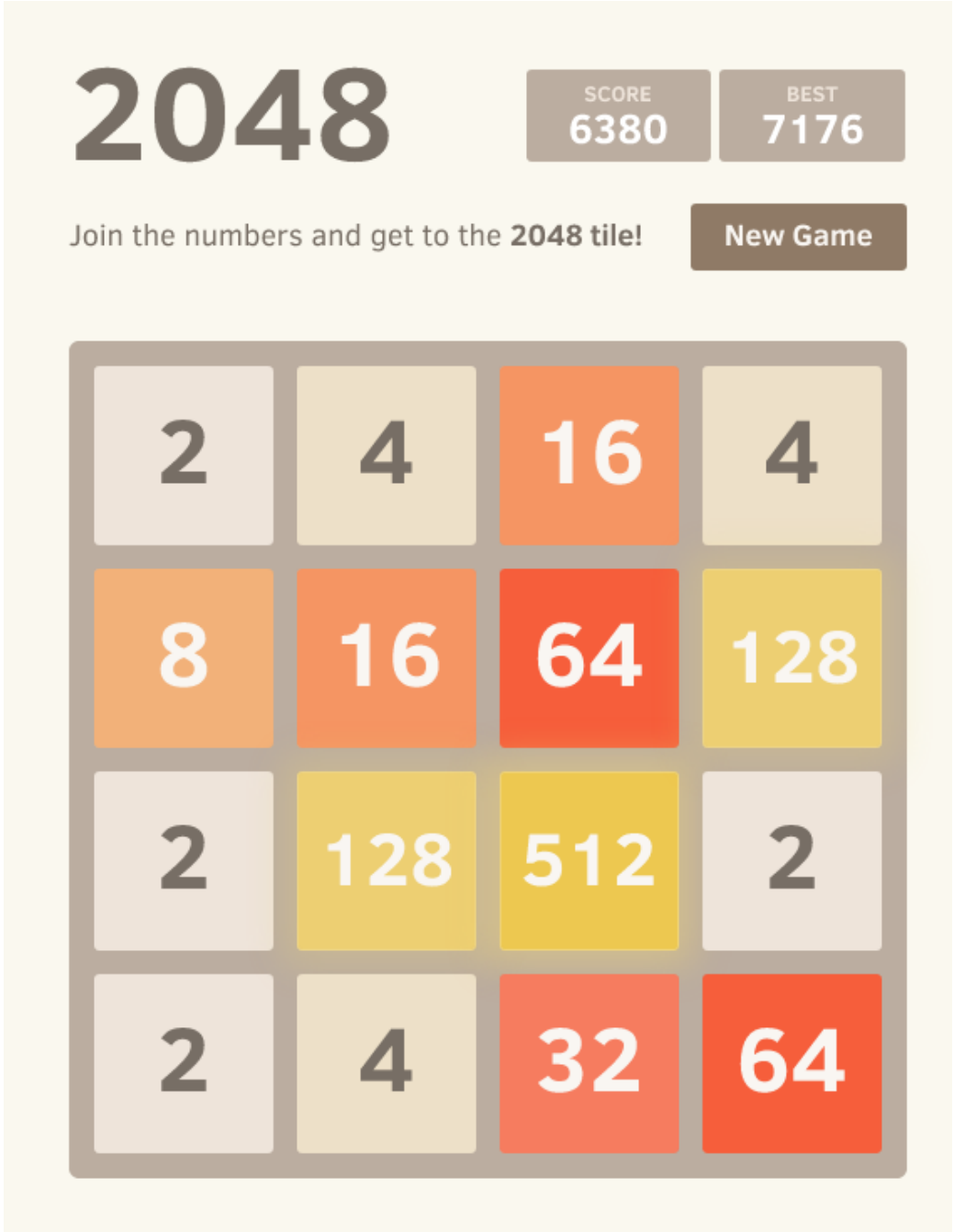
Cheryl tells the month to Albert and the day to Bernard.

Albert says, "I don't know the birthday, but I know Bernard doesn't know either."

Bernard then says, "I didn't know at first, but now I do know."

Albert then says, "Now I also know Cheryl's birthday."

When is Cheryl's birthday?

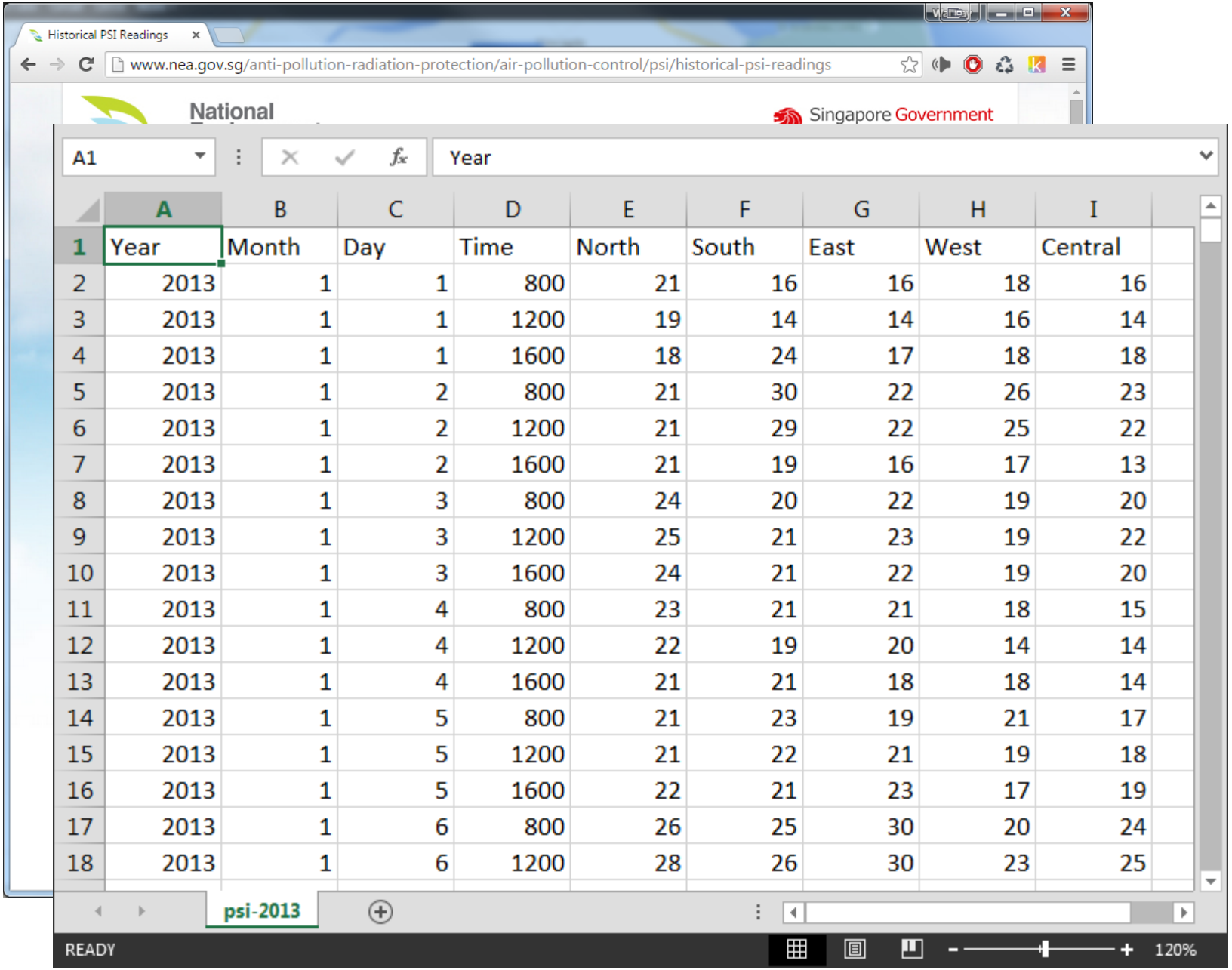

2048
SCORE
6380
BEST
7176
Join the numbers and get to the 2048 tile!
New Game
2 4 16 4
8 16 64 128
2 128 512 2
2 4 32 64

What will you use to

- Write a report?

- Do accounting?
- Statistical analysis?

- Find prime numbers?
- Solve a puzzle?
- Scrape data from a website?



	A	B	C	D	E	F	G	H	I
1	Year	Month	Day	Time	North	South	East	West	Central
2	2013	1	1	800	21	16	16	18	16
3	2013	1	1	1200	19	14	14	16	14
4	2013	1	1	1600	18	24	17	18	18
5	2013	1	2	800	21	30	22	26	23
6	2013	1	2	1200	21	29	22	25	22
7	2013	1	2	1600	21	19	16	17	13
8	2013	1	3	800	24	20	22	19	20
9	2013	1	3	1200	25	21	23	19	22
10	2013	1	3	1600	24	21	22	19	20
11	2013	1	4	800	23	21	21	18	15
12	2013	1	4	1200	22	19	20	14	14
13	2013	1	4	1600	21	21	18	18	14
14	2013	1	5	800	21	23	19	21	17
15	2013	1	5	1200	21	22	21	19	18
16	2013	1	5	1600	22	21	23	17	19
17	2013	1	6	800	26	25	30	20	24
18	2013	1	6	1200	28	26	30	23	25

What will you use to

- Write a report?

- Do accounting?
- Statistical analysis?

- Find prime numbers?
- Solve a puzzle?
- Scrape data from a website?

Do cool stuff

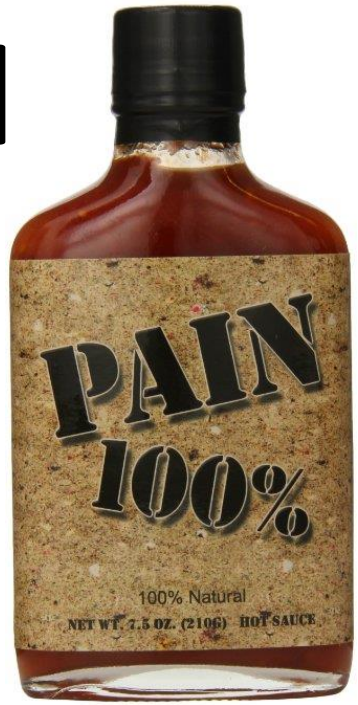
"You wouldn't believe what I have to do for my first task (at work)... Write some Python programme to automate some data processing. Hahahaha. Who would have thought!"

Mill Jianprasert
NUS B.B.A
Management Associate, Garena

Help simplify your work

Why should you take CS1010S

2. Because pain is good for you



WHAT

to expect in CS1010S

Work.

A lot of work.

A lot of
hard work.

A lot of
hard work.
and some fun :)

Lectures

Here!

Recitations

Start next week

Tutorials

Start in week 3

Tutorial ~~Balloting~~ *Allocation*

Do **NOT** use CORS

Fill in preferences in Survey

Make sure no clashes!

Recitation will continue to be balloted on CORS

Weekly Schedule

	Classes	Assignments	
Mon			
Tue		Lecture Training	
Wed	Lecture		Tutorial Training
Thu	Recitation		
Fri			
Sat			
Sun			
Mon	Tutorial		
Tue			

Important Dates

Wed 3 Oct: Midterm Test (6:30-8 pm)

Fri 19 Oct: Makeup Midterm (6:30 pm)

Sat 17 Nov: Practical Exam (12-6 pm)

Fri 23 Nov: Re-practical Exam (2 pm)

Please check and highlight any schedule clashes early

Google Calendar

https://calendar.google.com/calendar/ical/0eqa645n5o9cc8t1idjjlludss%40group.calendar.google.com/public/basic.ics

No textbook

Save money

The Art of Stealth Studying: How To Earn a 4.0 With Only 1.0 Hours of Work

Tips: Studying Study Hacks October 3rd. 2007, 1:40pm

Welcome new readers. *Study Hacks is a blog dedicated to productivity hacks for students. If you like this article, you might also like these posts on how as an MIT grad student I never work past 5 PM, the difference between work and pseudo-work, and the key to becoming both impressive and relaxed. If you prefer more technical advice, consider my article on using a Monotypic e-mail inbox, this survey of effective student time management techniques, this story of a student who got a 4.0 with 0.0 notes, and these instructions for building a paper research database.*

The Secret Art of Stealth Studying

Most students don't mind studying if the work gets done in focused chunks, spread out over a reasonable amount of time. For some, however, reasonable efficiency is not enough. They want more. They want to push academic productivity to its absolute limit. ***They want nothing less than to eliminate studying all together!***

I can't get you all the way to this goal. But I do know a technique that can get you close.

I'm a 31-year-old computer sci
interested in why some peopl
successful, enjoyable, meanin
while so many others do not.
geek, I'm not satisfied with sim
slogans (e.g., *"follow your pass*
conventional wisdom (e.g., *stu*
success requires stress). Instea
deeper, looking to decode un
patterns of success, in all thei
nuanced glory.

Subscribe to the Study Ha

Sponsor

The trick:
Consistency

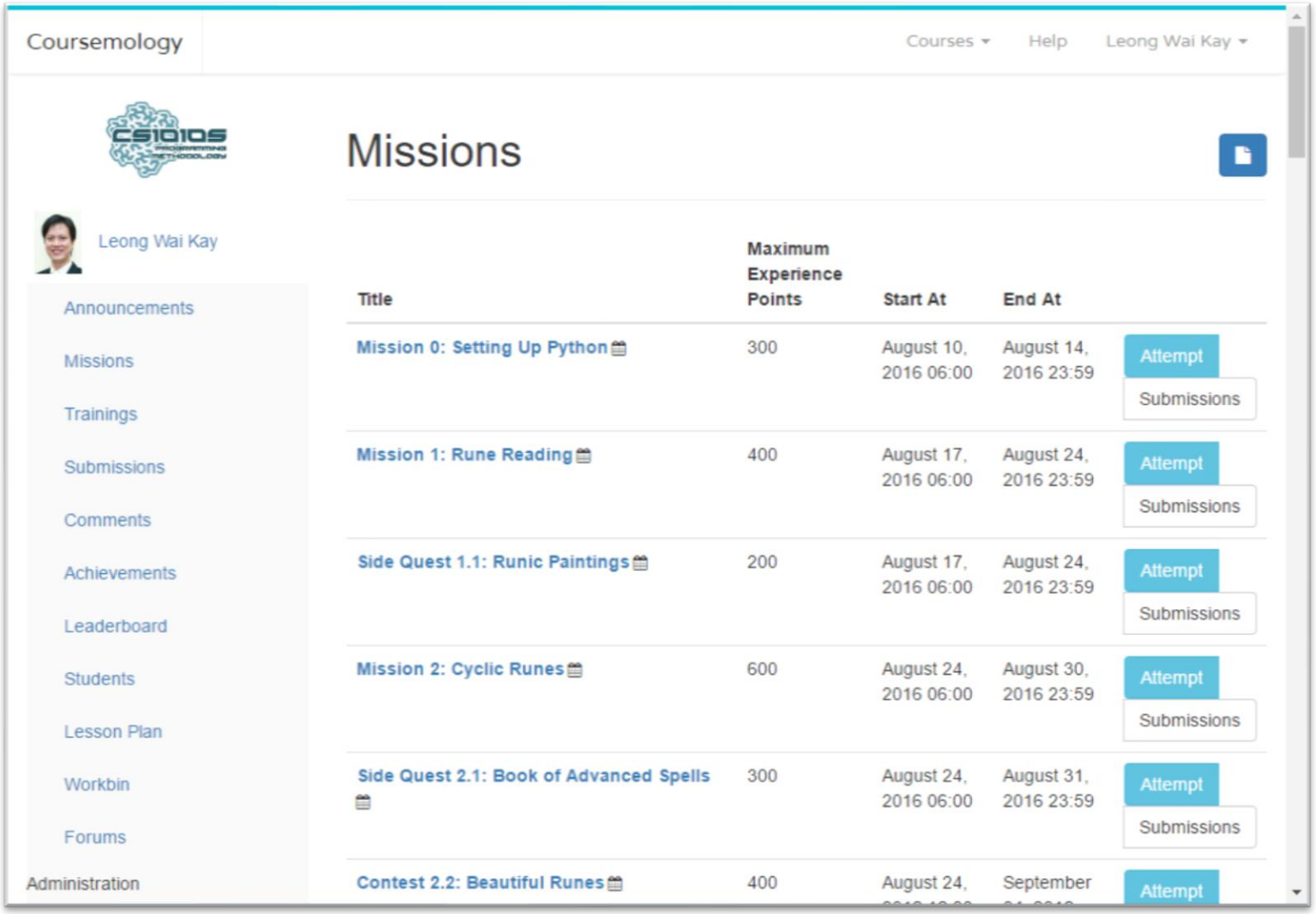
The problem:

Staying

consistent is

HARD


“gaming can make a
better world
- Jane McGonigal
17 Mar 2010



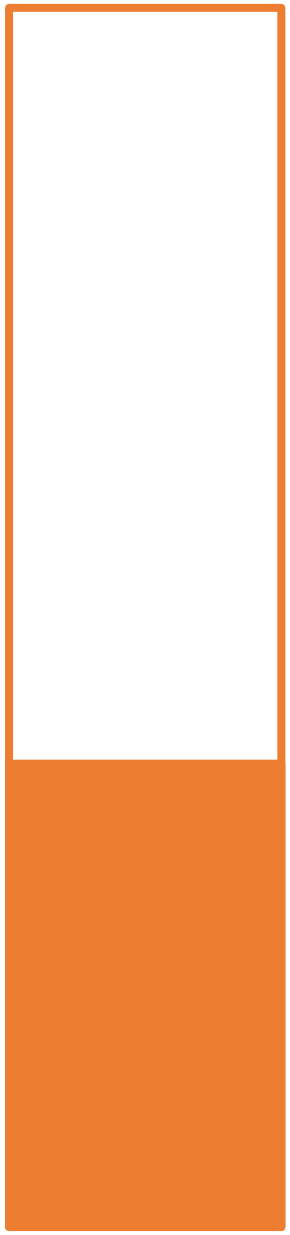
https://coursemology.org

7 Problem Sets

16 Main Missions

+ Side quests & Contests
+ 24/7 Grading
+ Unlock Achievements
+ Badges
+ Leaderboard

COMPLETE MISSION



+XP

COMPLETE
MISSION

+XP

COMPLETE MISSION

COMPLETE SIDEQUESTS

+XP

COMPLETE MISSION

WIN CONTESTS

COMPLETE SIDEQUESTS

+XP

COMPLETE MISSION

WIN CONTESTS

ATTEND TUTORIALS

COMPLETE SIDEQUESTS

+XP

COMPLETE MISSION

COMPLETE SIDEQUESTS

WIN CONTESTS

ATTEND TUTORIALS

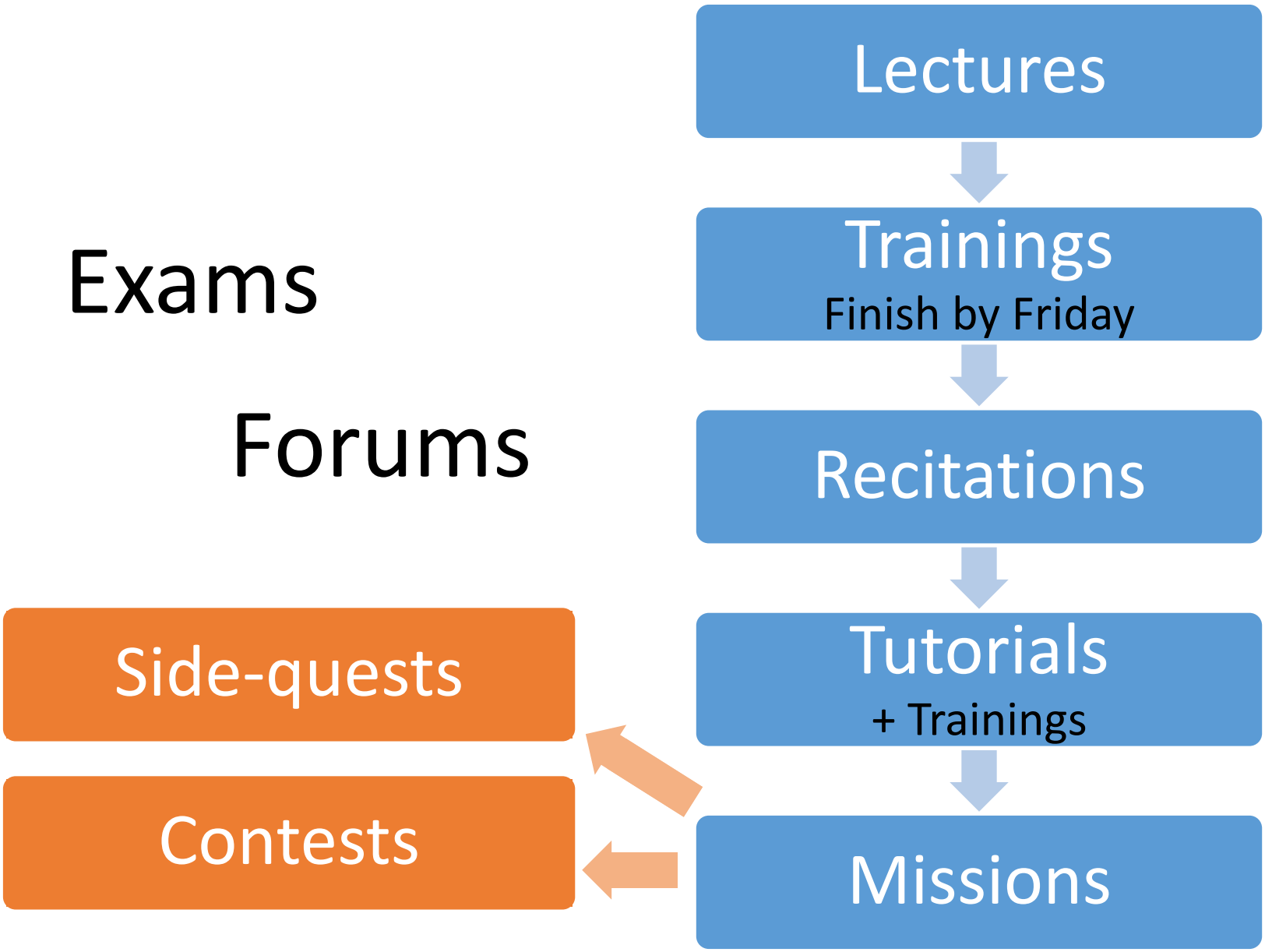
POST ON FORUM
(Reflections)

ENOUGH XP?

LEVEL UP!

Exams

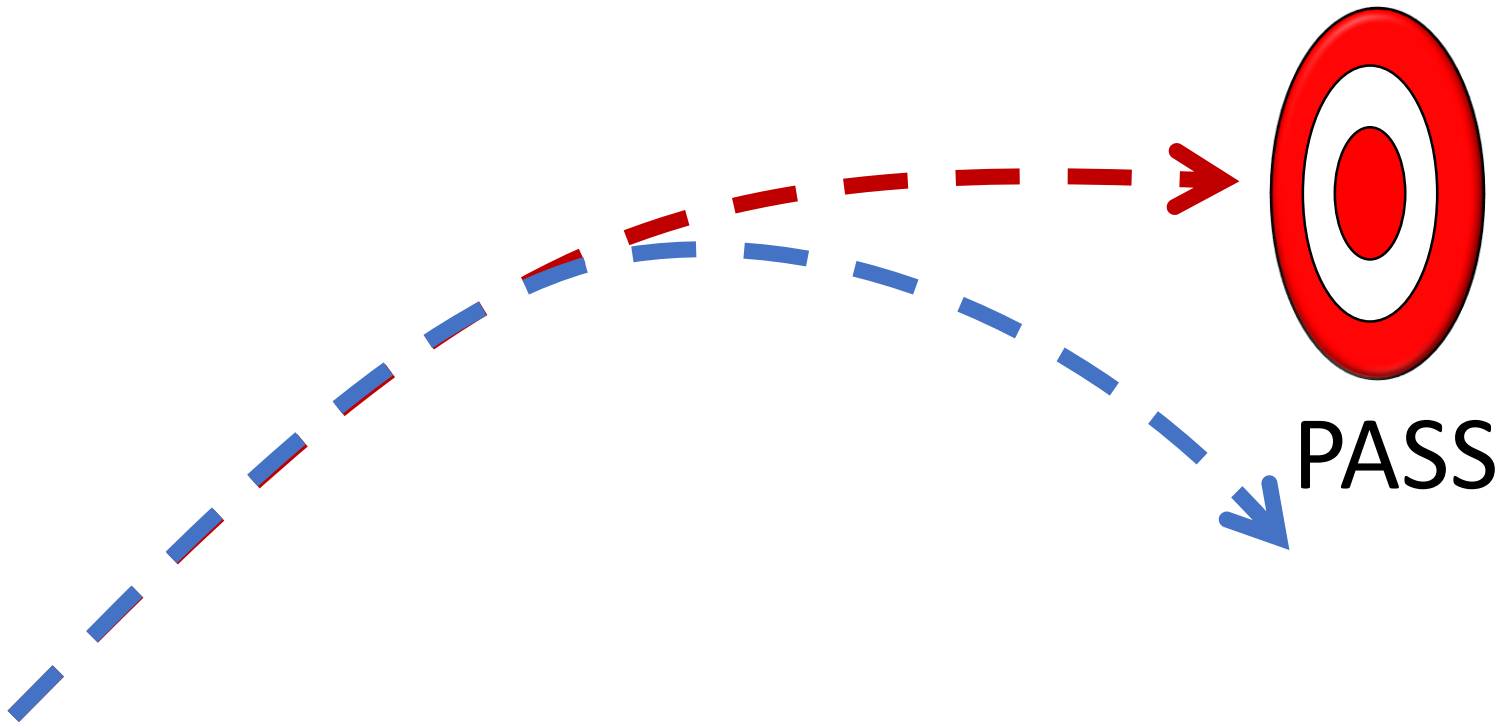
Forums



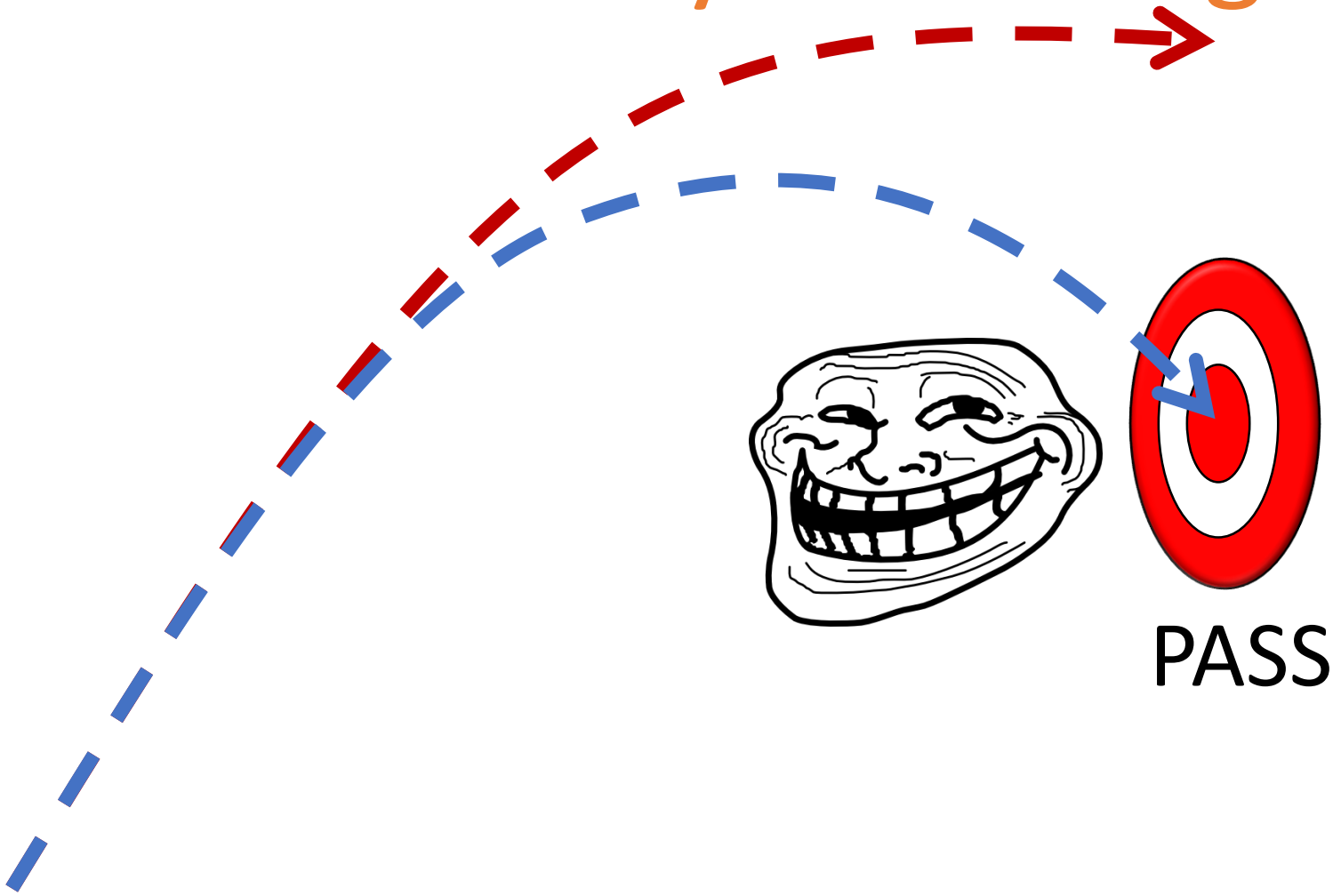
Assessment Overview

Coursemology	25%
Tutorial Participation	5%
Mid-term Exam	15%
Wed, 3 Oct, 6:30 – 8:00pm	
Practical Exam	15%
Sat, 17 Nov, 12:00 – 6:00pm	
Final Exam	40%
Tue, 27 Nov, 9:00 – 11:00am	

The Theory of Aiming



The Theory of Aiming



Not to worry.

Help is always available

Remedial

TBA

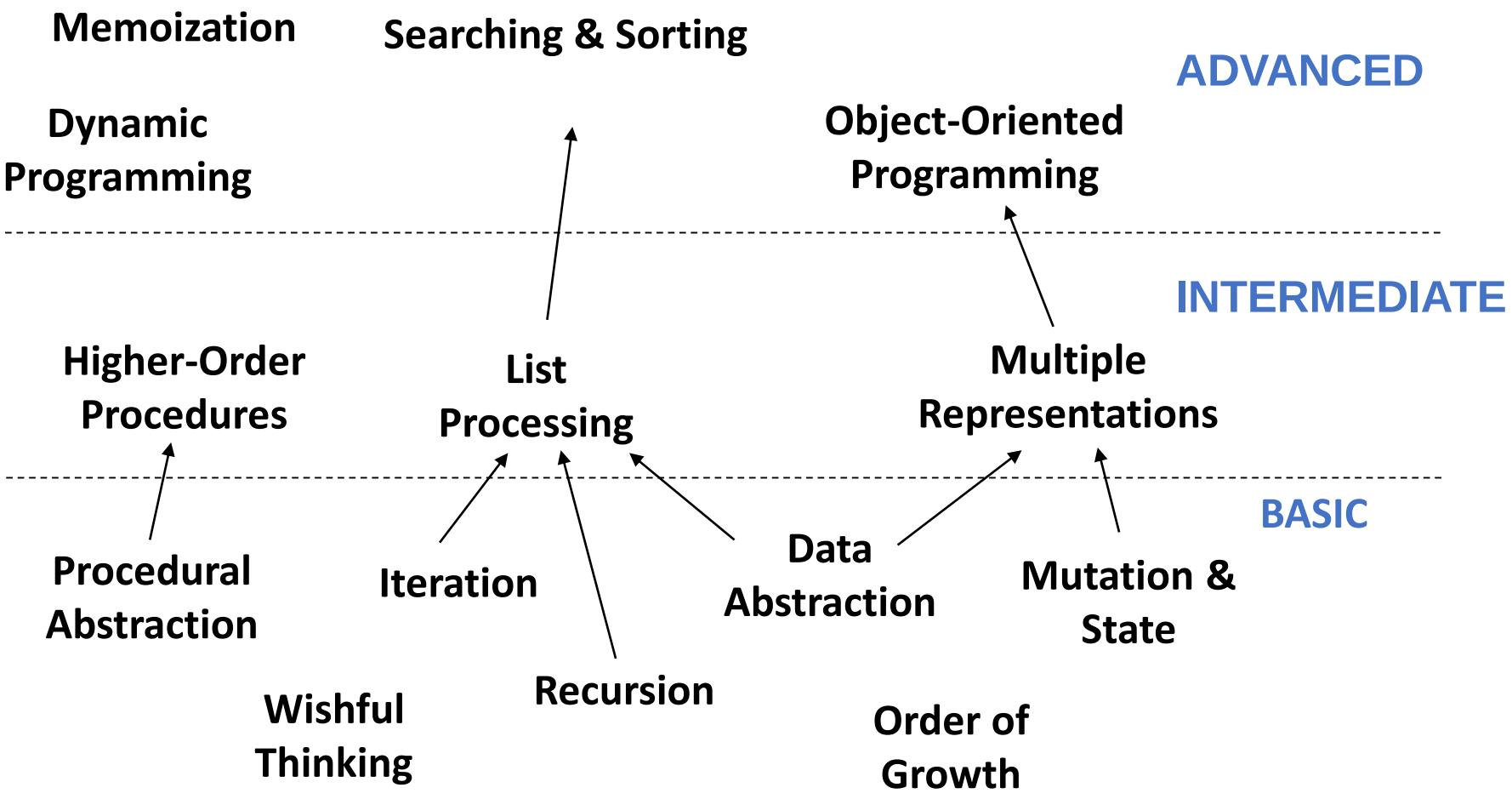
Email Us

cs1010s-staff@googlegroups.com

Discussion Forum

Do Subscribe

CS1010S Road Map



Fundamental **concepts** of computer programming

Bust some Myths

Myth 1

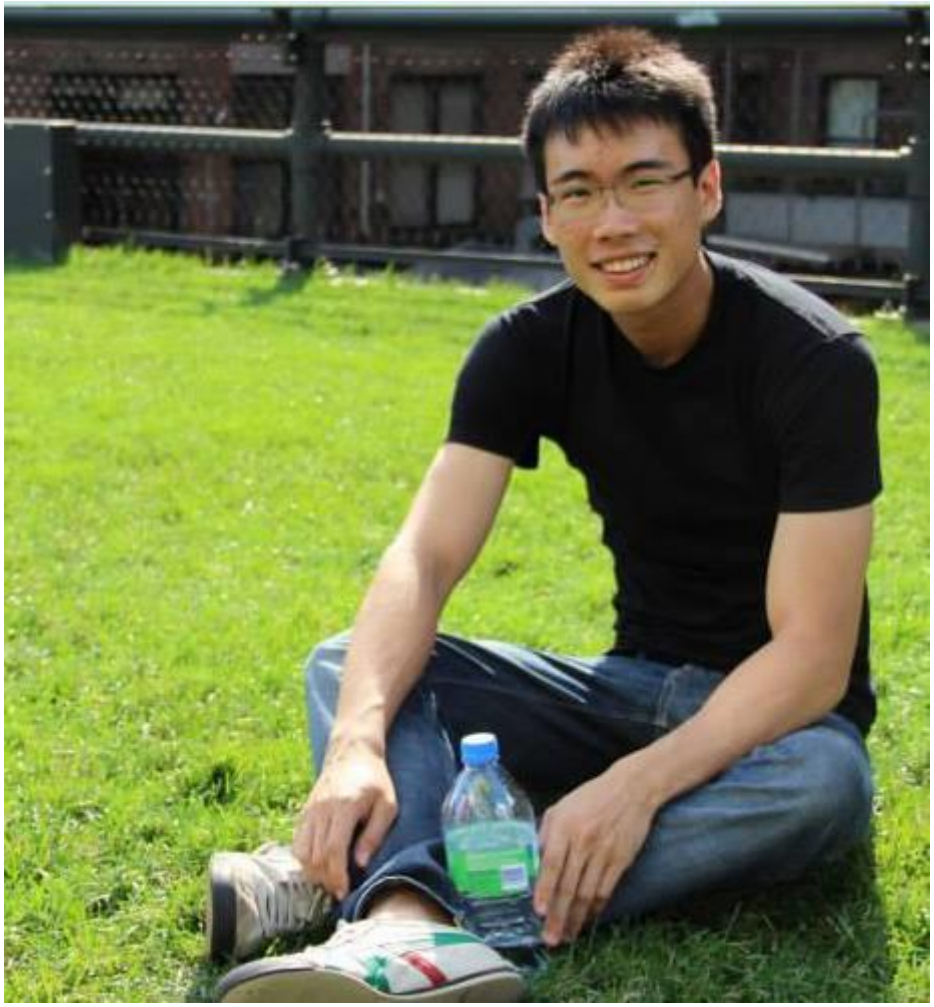
“Not everyone can learn to program”

Myth 1

"Not everyone can learn ~~to program~~ algebra"

Myth 2

“I have no programming experience. How to survive?”



Michael Yong

Year 2

Google Intern

USD7k a month

Myth 3

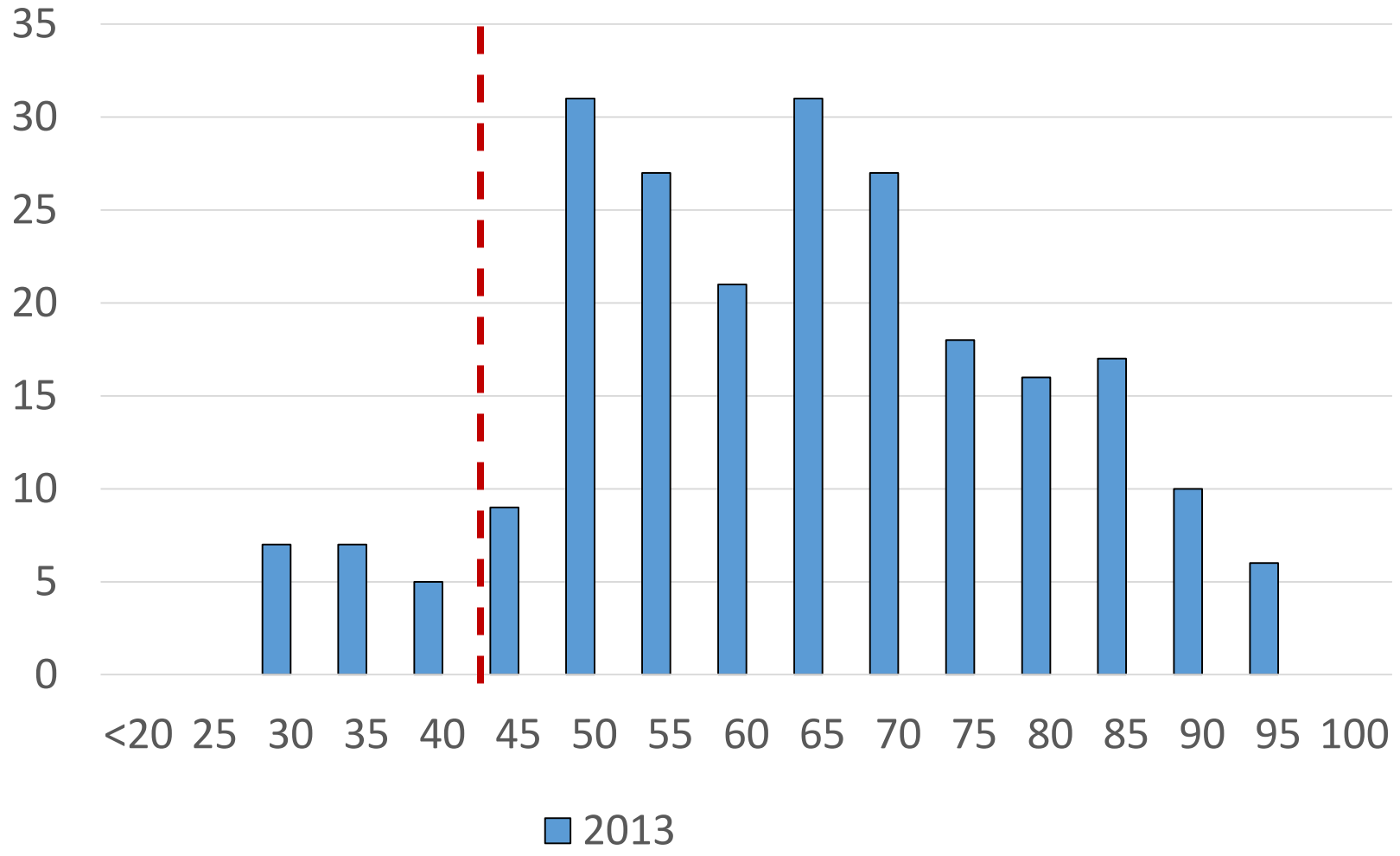
“I'm a girl, how can you expect me to program?”

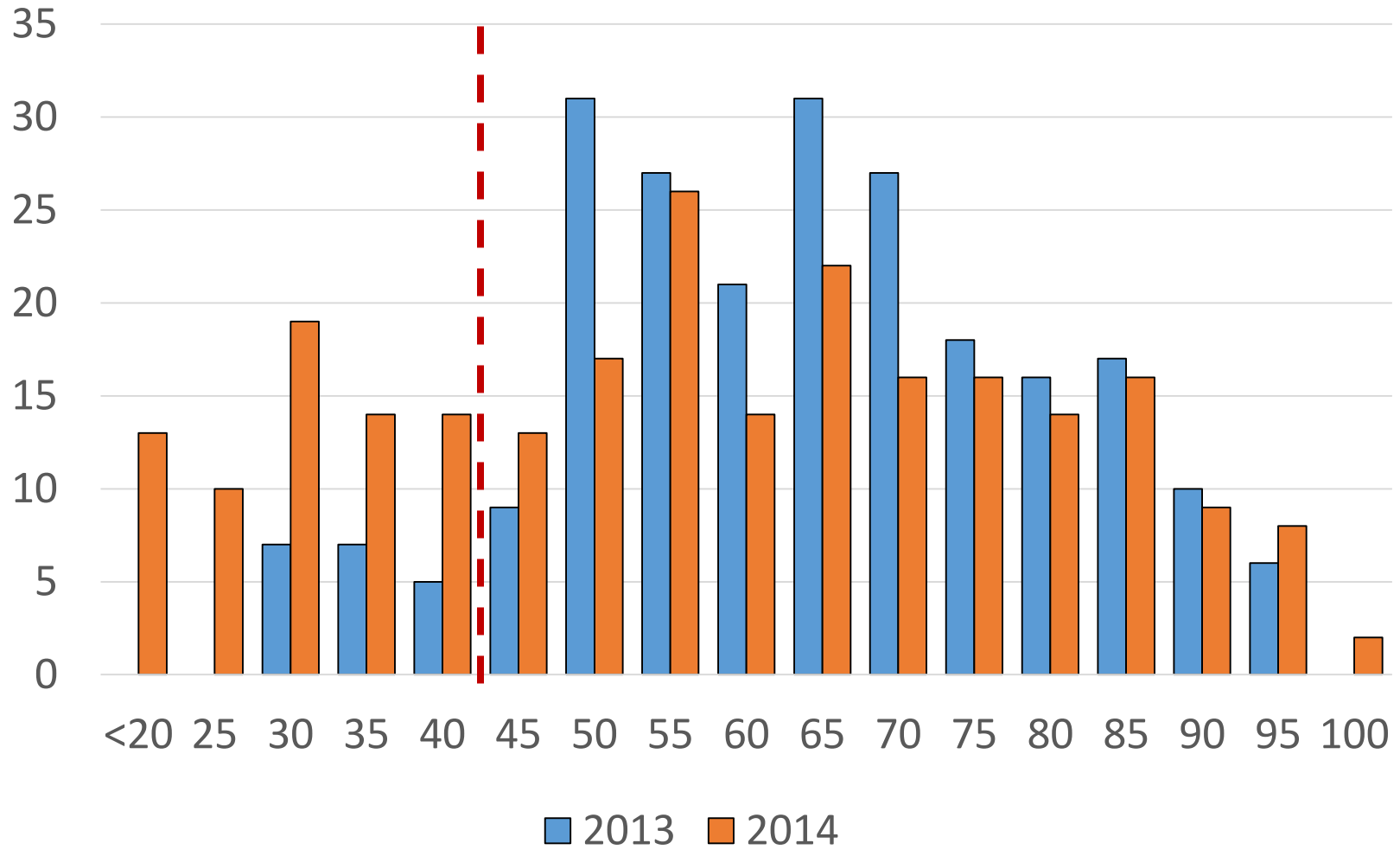
http://slatestarcodex.com/2017/08/07/contra-grant-on-exaggerated-differences/

Myth 4

“The bell-curve will save me.”

Grade-less

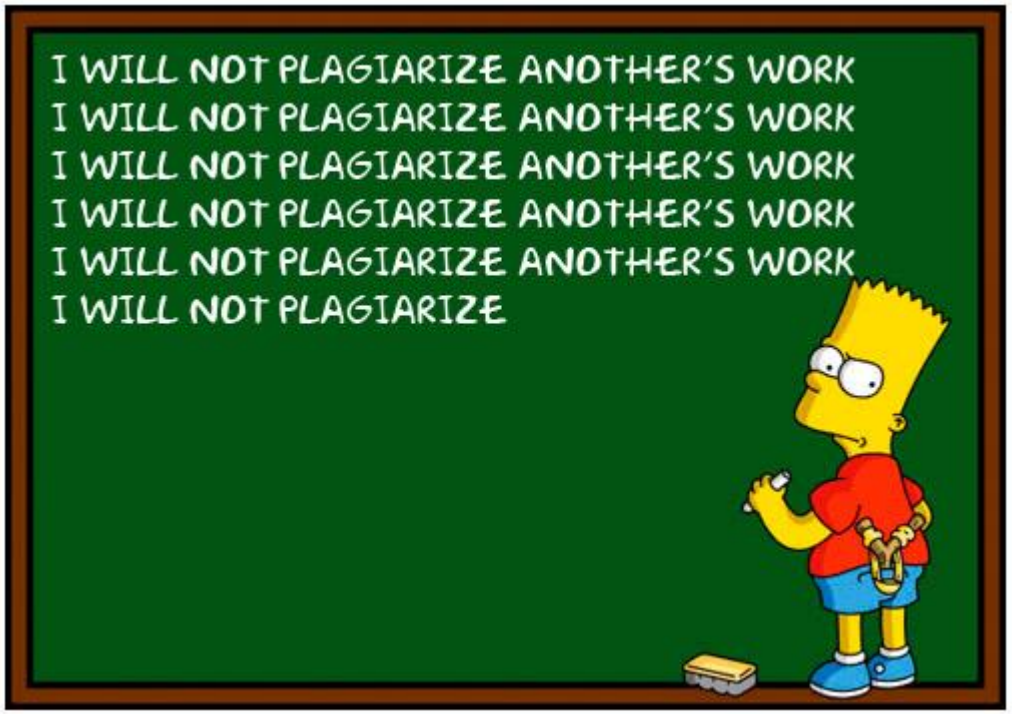

35
30
25
20
15
10
5
0
<20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
2013


35
30
25
20
15
10
5
0
<20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
2013
2014

Myth 5

“No one will bother if I copy my friend/senior’s code.”

Caught plagiarizing



1. Zero marks for affected Mission
2. Removal of S/U privilege

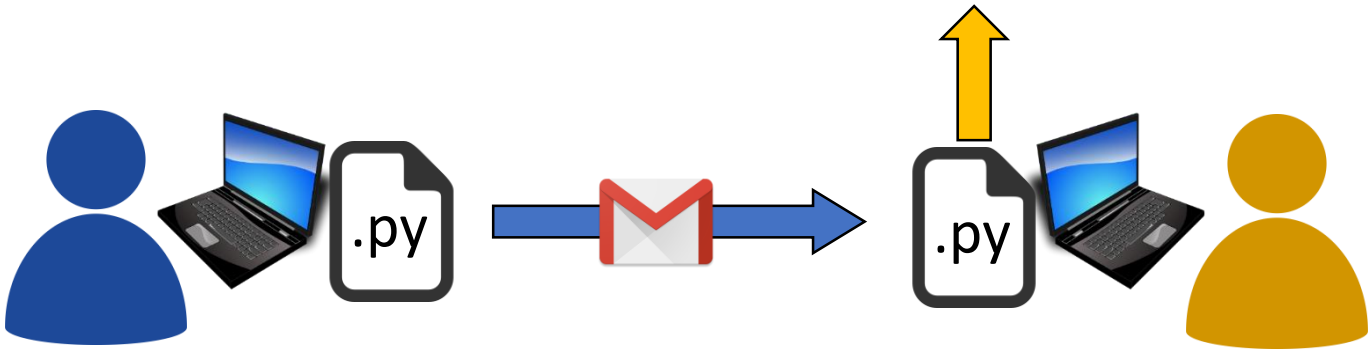
*Seniors sharing code will be considered to be abetting in plagiarism

Plagiarism

The act of presenting another's work or idea that as your own.

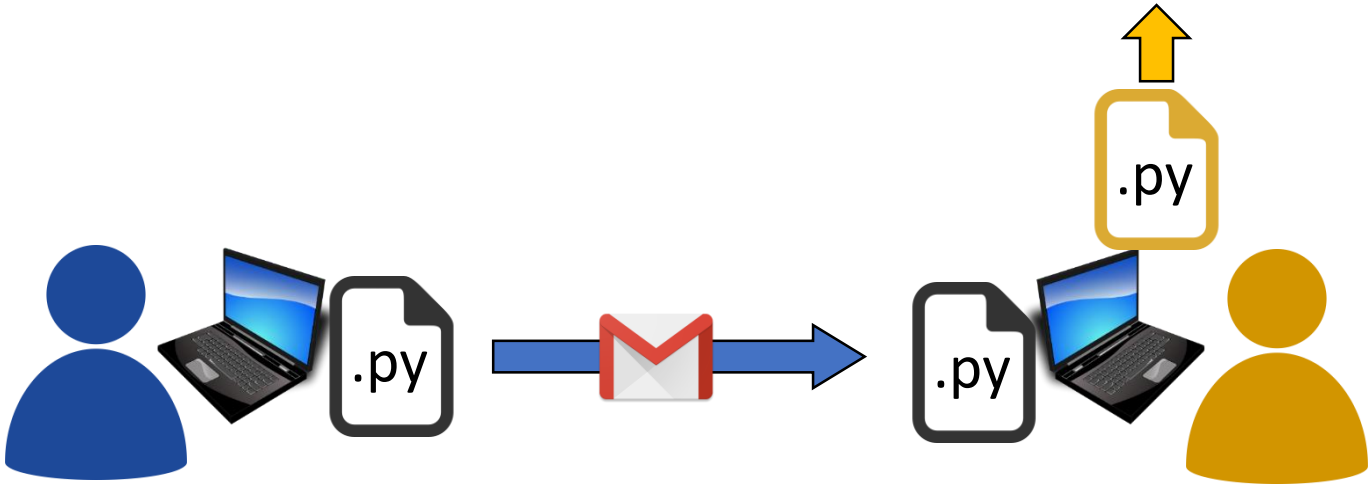
Plagiarism #1

Direct copying



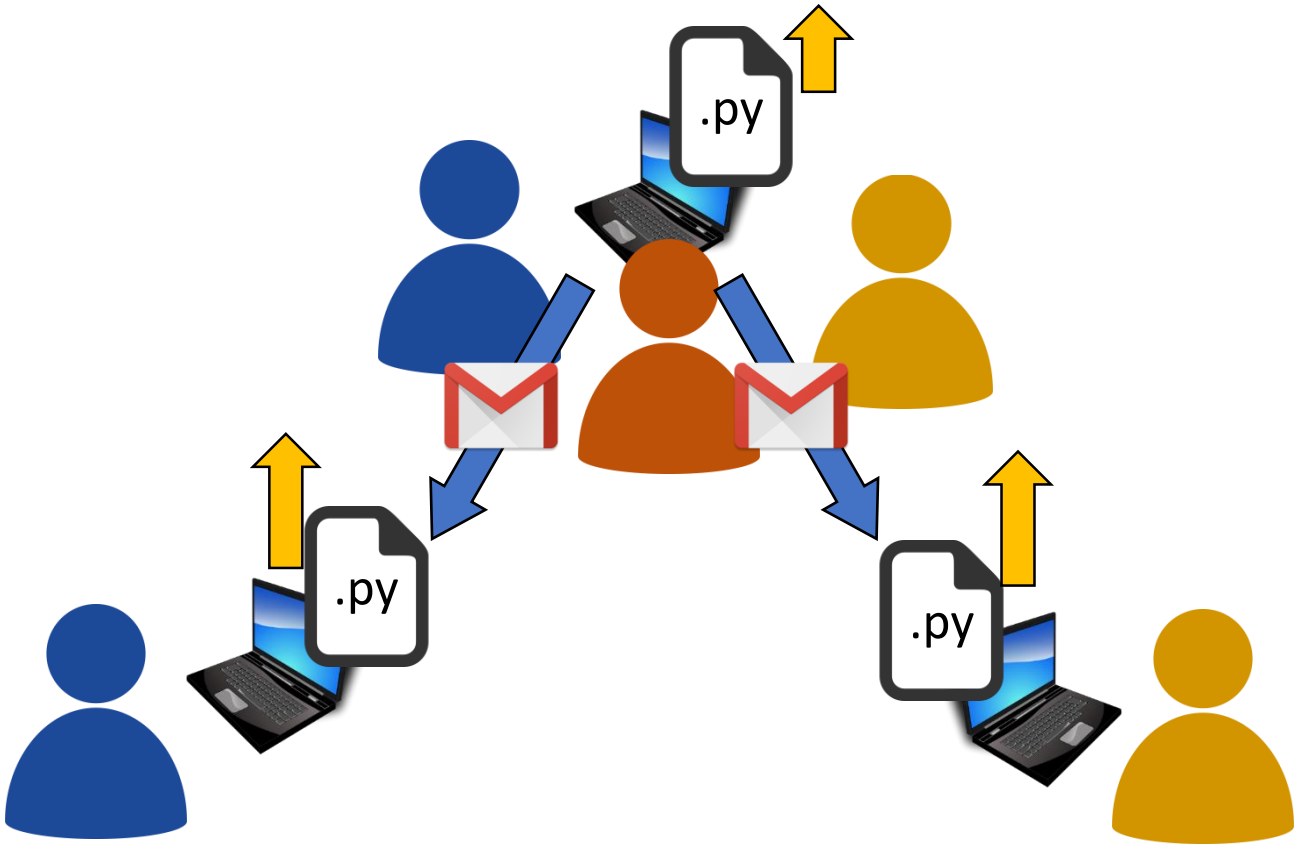
Plagiarism #2

"Referencing"



Plagiarism #3

"Discussion"

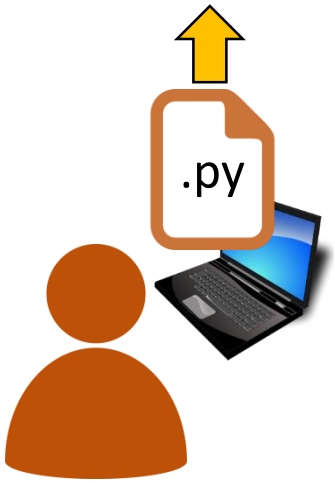


Not Plagiarism

Discussion (the proper way)

Not Plagiarism

Assistance



Why should you take CS1010S

3. Because coding is the new literacy

Well… not really

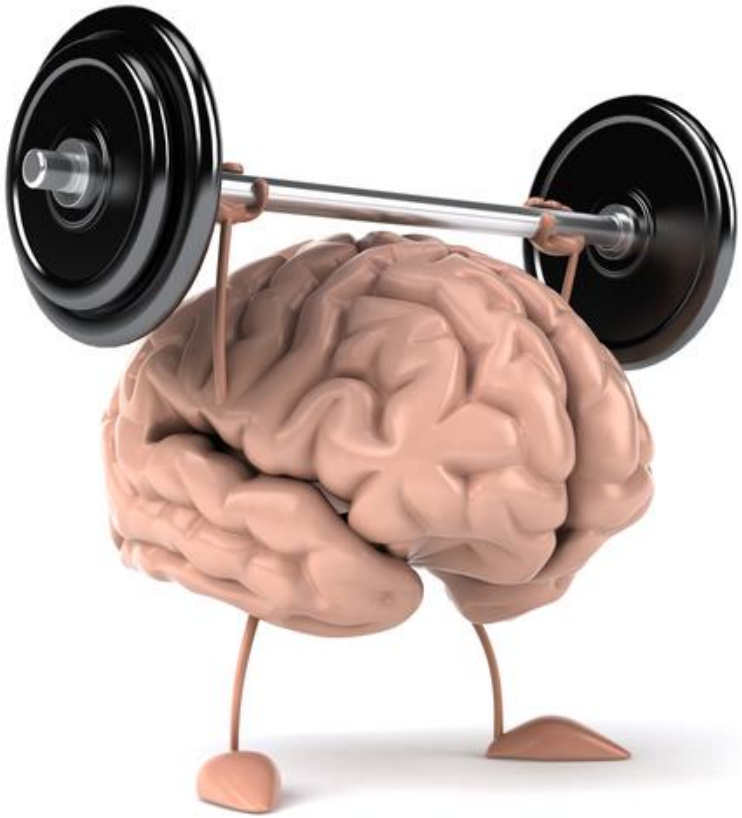
What are we teaching you?
What are you learning?

Python?

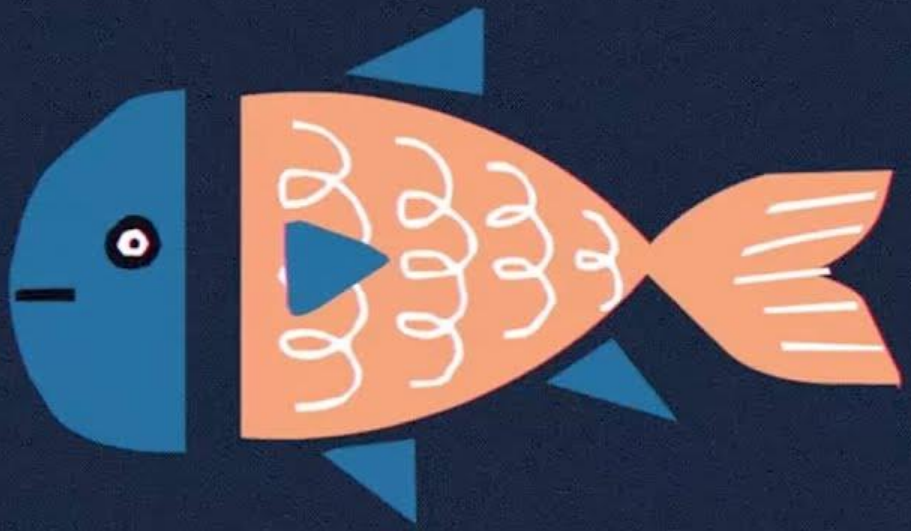
Not quite.

Python is just the tool

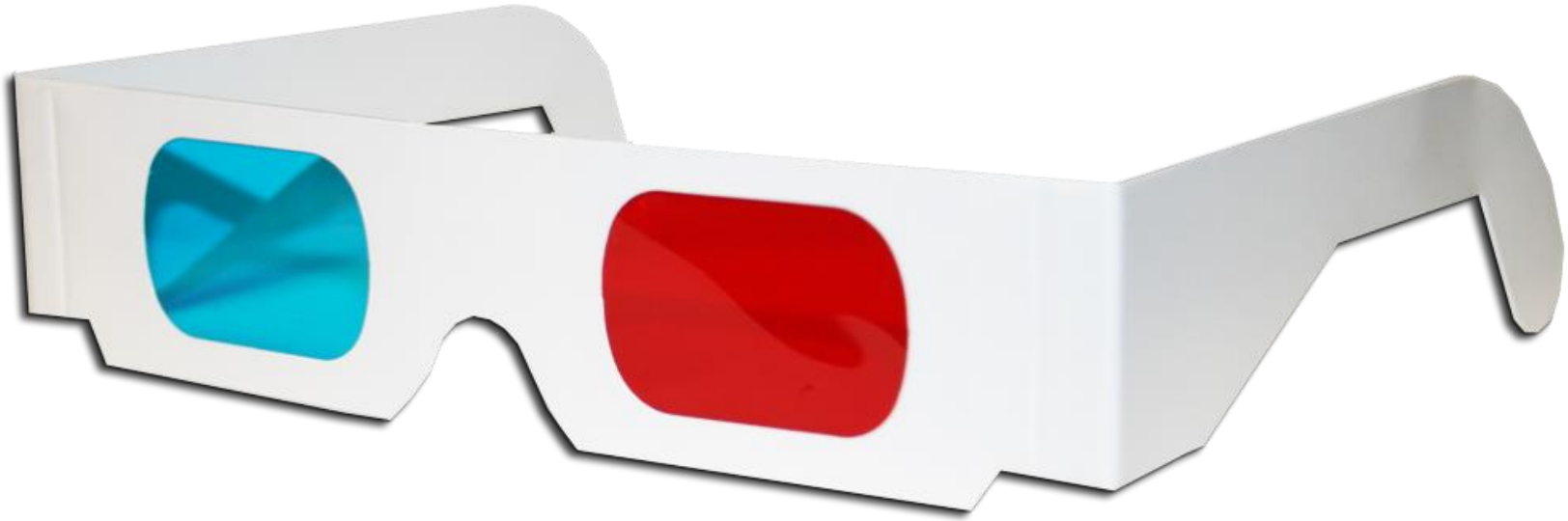
Critical Thinking
Problem Solving




It's the
process that
matters



CAN YOU SOLVE THE FISH RIDDLE?




3D

Let's hear it from your seniors

Initially I took CS1010S to learn programming, to help me to learn R for econometrics (I'm an Econs major planning to do a thesis). But I discovered that Python actually had libraries to do data work (pandas, numpy, dask, Stronne etc.), so I actually used these libraries for my Honours level module project. I found Python much easier to use than Stata, the program that the Econs department introduced us to. Stata was just a mess (no brackets for function calls, no assignment operators, error codes without helpful information (e.g. r(401) instead of, say, TypeError)).

I think that programming helps me automate and speed up repetitive tasks. I was surprised how much I was able to learn these libraries to deal with data, when I haven't even completed the module.

I also think that CS1010S had helped me to secure a data analytics internship at CPF. They wanted someone who had worked with real data before and some programming background. I was able to share with them my experience

Being the block committee in my hall this year, I had to plan an event for the entire block. Me & my partner decided to recreate the popular korean variety show, Running Man. At one stage of the game, participants have to collect numbers around the entire playing area & form a code. I blended python into the game by writing a code checker & my friends thought that it was cool to be able to use computing knowledge into daily lives. A few of my friends were motivated & are considering to take cs1010s in one of their semesters.

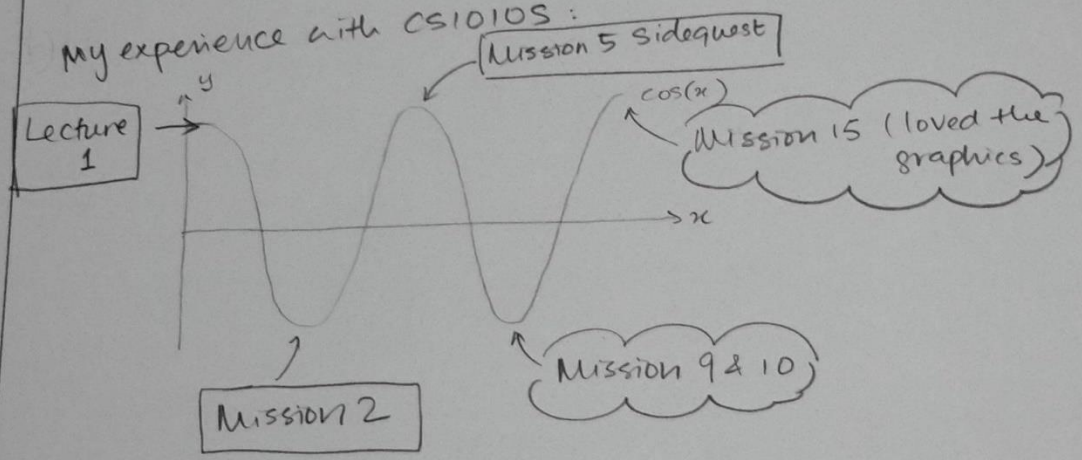
"How cool is that?"

4

If CS1010S won't set a student to learn time management, God knows what will.

That said, CS1010S is pretty fundamental to my major as Python is often used in Bioinformatics.

My experience with CS1010S:



Thank you all for the roller coaster ride!

4

(b) Not that interesting but I honestly have not been so stressed over a module. Not even during A lvls have I felt this stressed. When looking at mission 2, the one on runes, I actually cried or rather teared cos I just felt that there are too many things to do and I cannot handle. I jot down deadlines for the missions and I was like "what on earth". Yeah for the next few days (after ~~actually~~ actually teared LOL) I cried. Cos of other stuff also. I was rather depressed ~~for a fe~~ period. But yay... now I am here taking my finals ^‿^. I ~~ahh~~ have overcome it a ~~few~~ since like... mission 4 onwards ba. or Few weeks after CNY. But cs taught me I should be consistent w my work! Not consistent w/ handwriting though.

I learnt that determination is very important, it may seem like the missions and tasks are never ending, but look! It's the end! Before this module I never thought that tunnel vision would be a big problem for myself, but I really struggled at times, getting stuck on codes and missions. What helped was usually taking a break and letting my mind breathe a little to get perspective and tackle the question again. As much as I disliked the challenge and difficulty of the module, I secretly enjoyed parts of the process, because it really stirred me ~~to~~ out how far I can go if I'm willing to try. Okay I have no time for the other questions, so this is it. ☺

I have learnt computational thinking, of how to break down problems step by step to achieve my goals in life. I have learnt to be creative in resolving problems, to always try again and never give up, and to see a problem differently each time you fail.

I have also made friends during my course in CS1010S. And I believe that friendship through hardship and make the hardships easier. can last

I found a partner through Mission 11. That's a story about my experience in CS1010S this sem.


many paths to
success

HOW
to learn Python
(In 15 mins)

IDLE

http://www.python.org

Python 2

vs

Python 3

Python 3.7.0
for CS1010S

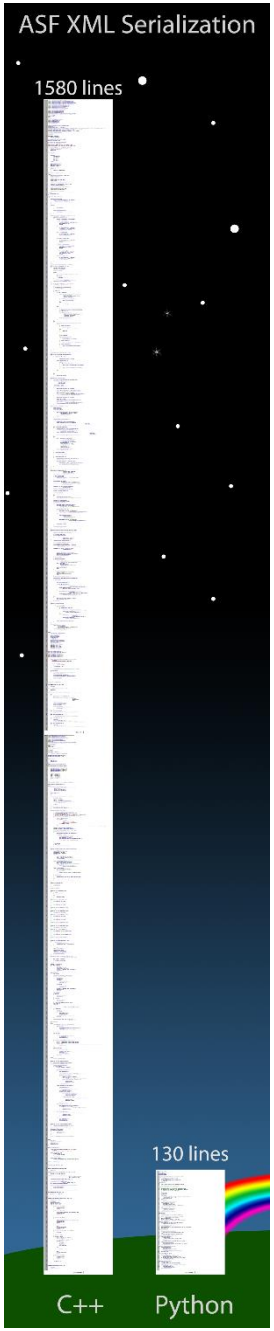
Why Python

- Clear and readable syntax
- Intuitive
- Natural expression
- Powerful
- Popular & Relevant

Example: PayPal

- ASF XML Serialization
  - C++ → 1580 lines
  - Python → 130 lines

https://www.paypal-engineering.com/2014/12/10/10-myths-of-enterprise-python/



Simplicity

Futile to Teach a Language

Here today, gone tomorrow

Computational Thinking

What is a program?

- A sequence of instructions
- That modifies some state (of the computer)

What is a program?

Example:

1. Pick up the crate
2. Move 5 meters
3. Turn right
4. Move 5 meters
5. Put down the crate

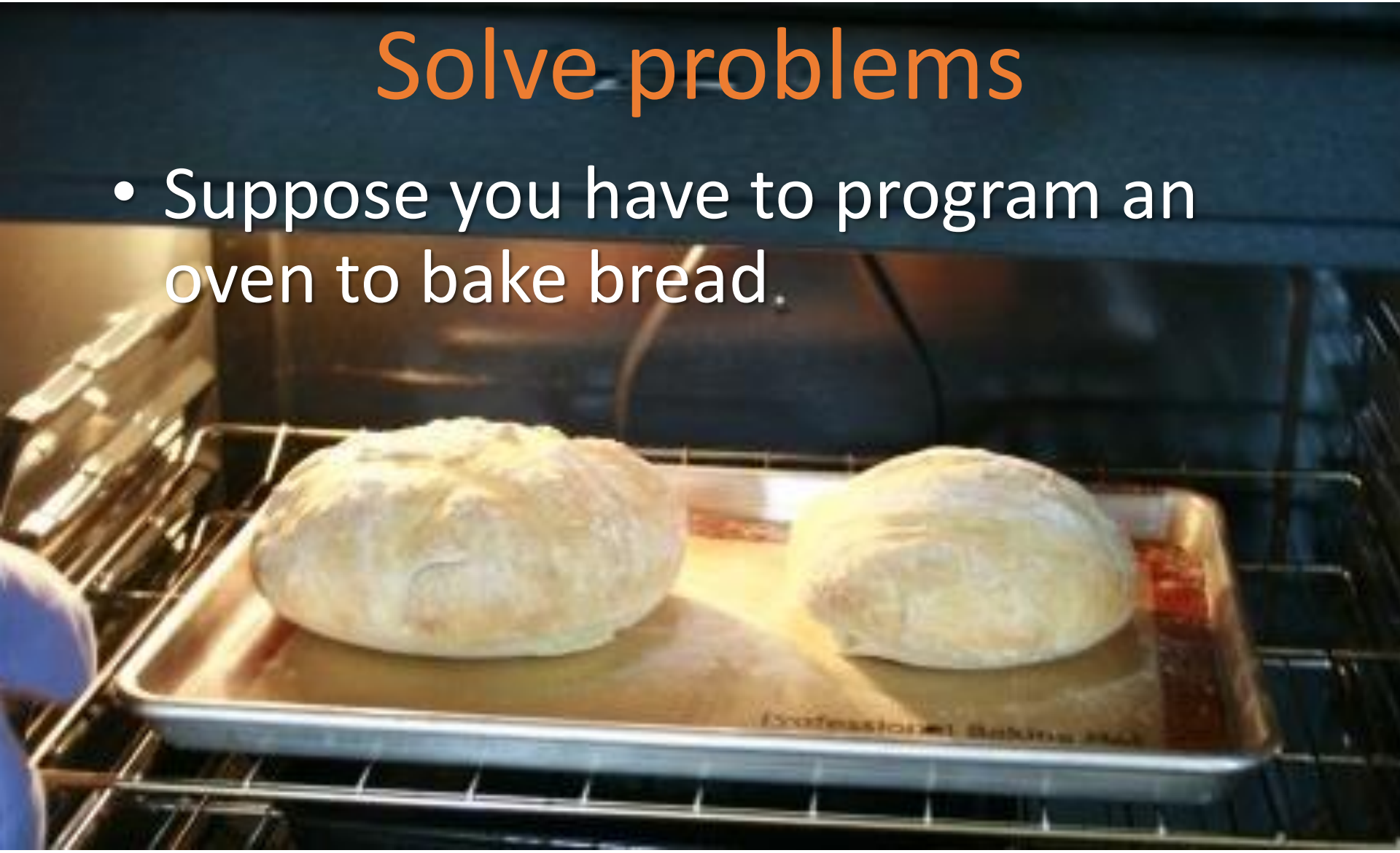
What is the state being modified?

Python program

```
a = 1
b = 2
c = a + b
if c < 0:
    print('Yes')
else:
    print('No')
```

Solve problems

- Suppose you have to program an oven to bake bread.



Idea

- Bake at 375°F for 30-35 minutes or until golden brown and bread sounds hollow when tapped.

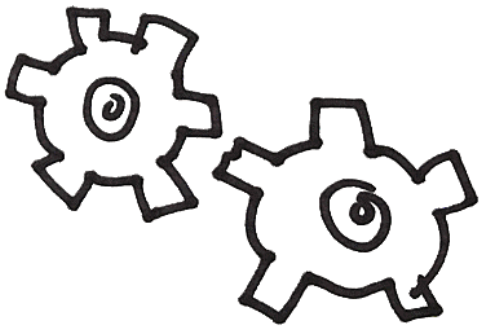
Implementation?

1. Start timer
2. Measure temperature
3. If temp < 375°F then turn on heating element
4. If temp > 375°F then turn off heating element
5. If timer < 35 mins go back to step 2

From IDEA

to

IMPLEMENTATION



Elements of Programming

1. Primitives

- Numbers:
  `4 , 7/2 , 428.3`
- Operators:
  `+ , - , * , /`
- Symbols:
  `a , pi , foo`

2. Means of Combination

```
5 + 3
```
8

```
((5 + 3)-(2 * 3))
```
2

3. Means of Abstraction

```
7 + 6
```

13

```
a = 3
```

```
a + 6
```

9

4. Controlling Logic

If the light is red,

    stop.

Else if it is amber,

    accelerate.

Else,

    proceed with caution.

While you are hungry,

eat a spoonful of rice.

Burp.

Variables

- Start with 'a'-'z' or 'A'-'Z' or '_'
- Contain only alphanumeric characters or '_'
- Case sensitive

```
Ian_Lee != ian_lee
```

Variables

- Avoid reserved keywords e.g. `if`
- Python convention: lower case letters separated by '_'
  - e.g. `count_change`
- Store memory addresses

Types

int	`8  45  1234`
float	`2.3  3.14159`
bool	`True False`
str	`"cs1010s"`
	`'cs1010s'`
None	

Type(…)

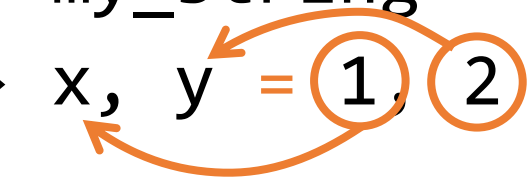
```
>>> type(123)
<class 'int'>

>>> type('123')
<class 'str'>

>>> type(None)
<class 'None'>
```

Type conversion

```
>>> str(123)
'123'

>>> float('45.2')
45.2

>>> int(23.8)
23

>>> int('cs1010s')
ValueError!
```

Assignment

```
>>> abc = 18
>>> my_string = 'This is my string'
>>> x, y = 1, 2
```



variable	address
abc	*id1*
my_string	*id2*
x	*id3*
y	*id4*

id1: int
18

id2: str
“This is my string”

id3: int
1

id4: int
2

Operators

Arithmetic: + - * / ** // %

```
>>> a = 2 * 3
>>> a
6

>>> 2 ** 3
8
```

Operators

Arithmetic: + - * / ** // %

```
>>> 11 / 3
3.6666666666666665

>>> 11 // 3
3

>>> 11 % 3
2
```

Truth Values

Statements can be either true or false

- $2 > 1$ is true
- $5 < 3$ is false

Operators

Comparison: `>` `>=` `<` `<=` `==` `!=`

```
>>> 1 <= 10
True

>>> 5 > 15
False

>>> 5 <= 5
True
```

Operators

Comparison: `>` `>=` `<` `<=` `==` `!=`

```
>>> 2 != 3
True

>>> '1' == 1
False

>>> False == False
True
```

Operators

Logic: `or` `and` `not`

```
>>> True or False
True

>>> True and False
False

>>> not False
True
```

Operators

Logic: `or` `and` `not`

a or b	True if either a or b is True
a and b	True if both a and b are True
not a	True if a is not True

Truth Tables

OR	True	False
True	True	True
False	True	False

AND	True	False
True	True	False
False	False	False

NOT	
True	False
False	True

Truth Value Revisted

- Python has keywords `True` and `False`
- In Python 3.x, `True` and `False` will be equal to `1` and `0`
- Anything that is not `0` or `empty` will be evaluated as `True`

Operators

Logic: `or` `and` `not`

```
>>> True and 0
0

>>> not 'abc'
False

>>> 1 or 0
1
```

Strings

```
>>> s = 'ba'
>>> t = 'ck'
>>> s + t
'back'

>>> t = s + 'na' * 2
>>> t
'banana'
```

```
>>> 'z' in t
False

>>> 'bananb' > t
True

>>> 'banan' <= t
True

>>> 'c' < t
False
```

String Slicing

- A String is a sequence of characters
- We can index a string, i.e.

```
>>> s = 'abcd'
>>> s[0]
'a'
>>> s[2]
'c'
```

- First character is `0`

String Slicing

```
s[start:stop:step]
```

non-inclusive

```
>>> s = 'abcdef'
>>> s[0:2]
'ab'
>>> s[1:2]
'b'
>>> s[:2]
'ab'
```

```
>>> s[1:5:3]
'be'
>>> s[::2]
'ace'
```

Conditional

```
if <expr>:
    statement(s)
```

e.g.

```
>>> a = 3
>>> if a > 0:
        print('Good')
```

indentation

```
'Good'
```

Conditional

```
if <expr>:
    statement(s)
else:
    statement(s)
```

e.g.

```
>>> a = 3
>>> if a > 0:
        print('yes')
    else:
        print('no')

'yes'
```

Conditional

```
if <expr>:
    statement(s)
elif <expr>:
    statements(s)
else:
    statement(s)
```

e.g.

```
>>> a = -3
>>> if a > 0:
        print('yes')
elif a == 0:
        print('no')
else:
        print('huh')

'huh'
```

Conditional

```
if <expr>:
    statement(s)
elif <expr>:
    statements(s)
elif <expr>:
    statements(s)
else:
    statement(s)
```

e.g.

```
>>> a = 4
>>> if a > 0:
        print('yes')
elif a == 0:
        print('no')
elif a == 4:
        print('ahh')
else:
        print('huh')

'yes'
```

Conditional

```
while <expr>:
    statement(s)
```

```
>>> a = 0
>>> while a < 5:
        a = a + 1
        print(a)

1
2
3
4
5
```

Python
in 15 mins

Am I now a programmer?

What does it take to be a programmer?

Do you know

1. your “ABC”?
2. how to form words?
3. how to write sentences?

If you answered YES to all,

surely you must be a poet

What does it take to be a poet?

What does it take to be a programmer?

To be programmer

1. Know the elements of programming
2. Know the grammar of programs
3. Able to read and understand programs
4. Must write working programs

This step is not trivial

Learning Outcomes

At the end of the course, you should be able to:

- describe the output of a basic program
- write a basic program to solve a given problem

Practice

Practice

Practice

Learning how to Learn

Feeling Overwhelmed?

Questions?

Ways to earn XP

Survey +100 XP

before 21 Aug

Training +100 XP

By Friday

+ 50 XP

otherwise

See you
next week.