CS1010S Programming Methodology

Lecture 1

Introduction to CS1010S & Python

11 Jan 2023

Three Questions

Why should you take CS1010S?

What to expect in CS1010S?

How to learn Python?

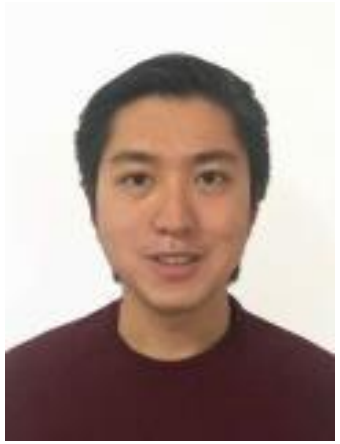
The Teaching Staff

Ashish

Waikay

Nitya

Kelvin



Tutors







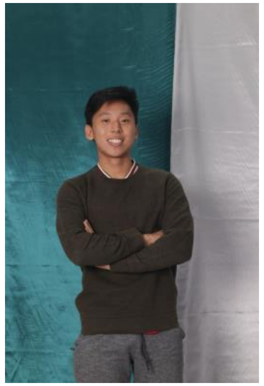




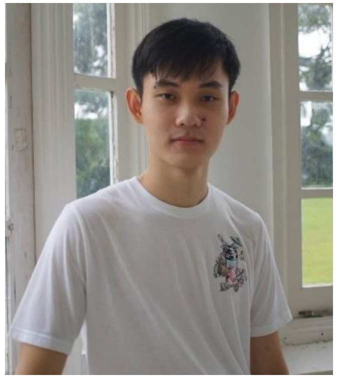















People who helped make CS1010S what it is!

Prof Ben Leong

Prof Ket Fah

Prof Alan

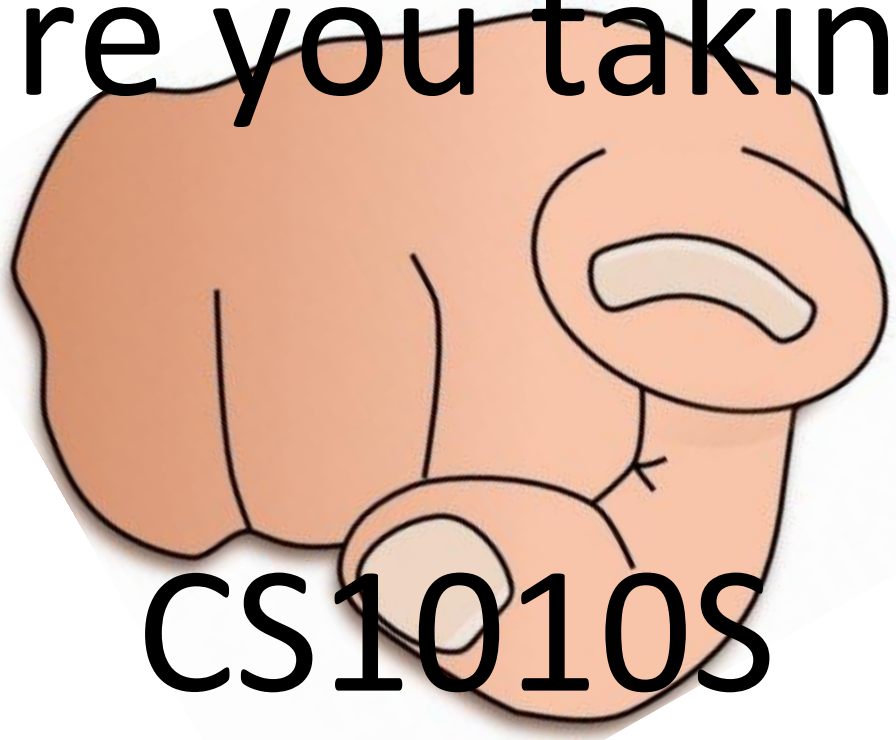
Prof Lifeng

WHY

should you take CS1010S?

WHY

are you taking



CS1010S

Who is taking CS1010S?

What is your major or program or home department?

BBA

Science (DSA, Bioinformatics, BA, Math, Econ)

Science (others)

Pharmacy

FASS

CHS

CDE

Others

WHY are you taking CS1010S?

Because I love programming.

Because I want to learn programming.

Because I am interested in computing.

Core Requirement!

WHY should you take CS1010S?

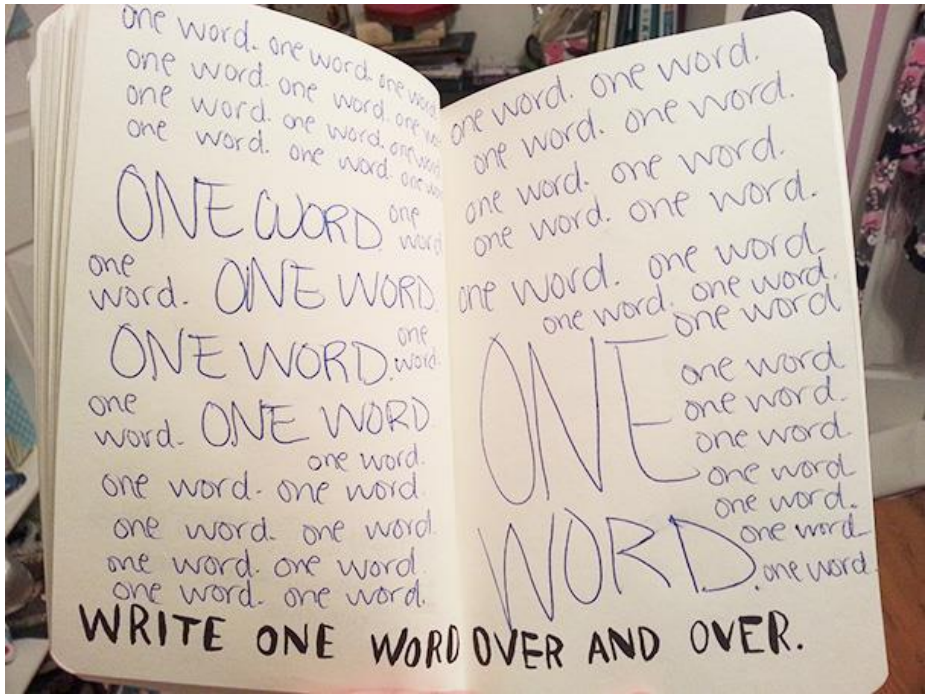
Computers are very useful!

- Writing reports. Word, Notepad, LaTeX, Grammarly, …
- Playing with numbers. Excel, spreadsheet, …
- Data visualization. Excel, Tableau, Origin, …
- Scientific computing. Matlab, Octave, …

But they are equally dumb!

They need to be given precise instructions.

And they will execute every instruction given to them; even the bad one!



WHAT to expect in CS1010S?

CS1010S vs COS2000

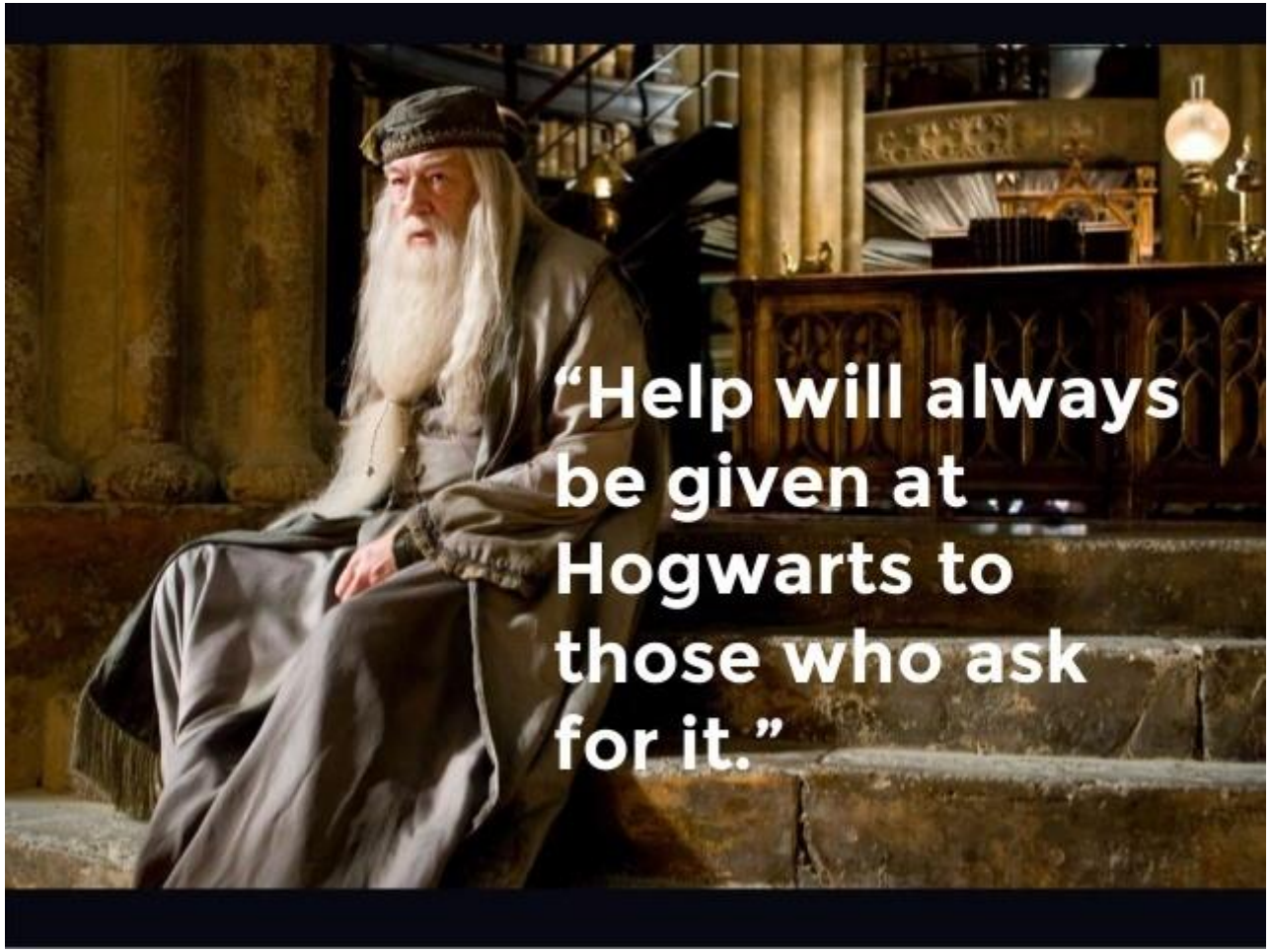
Culinary Degree vs Cooking Class

WHAT to expect in CS1010S?

- Work.
- Some more work…
- At times - a lot of hard work!

But all of this can be fun!

Not to worry!



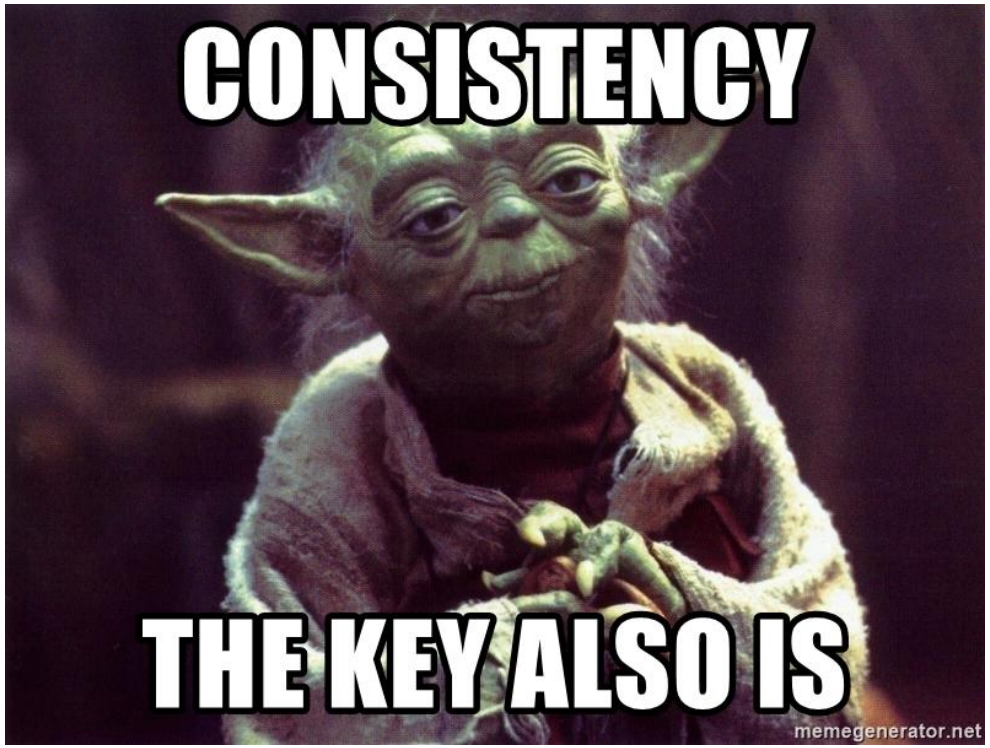
Three Pillars of CS1010S

- Lectures

- Recitations
  - Start in the next week

- Tutorials
  - Start in the Week 3

How to excel in CS1010S?



coursemology

Courses Raymond Tang

Raymond Tang

ACHIEVEMENT 0

LEVEL 7

1810 / 1900 EXP

- Announcements
- Missions
- Trainings
- Submissions
- Achievements
- Leaderboard
- Students
- Comments

CS1010S - Programming Methodology

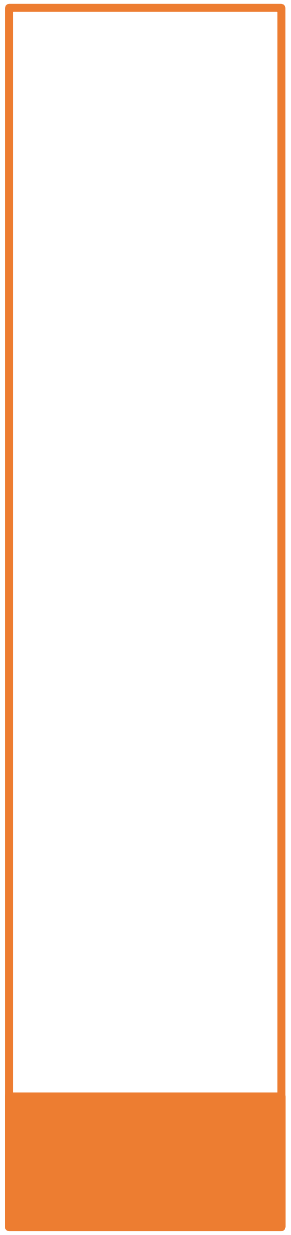
Missions

» Filter (click to reveal)

MISSION	MAX EXP	START DATE	END DATE
Mission 0: Setting Up Python	400	14-08-2013	17-08-2013
Mission 1: Rune Reading	500	21-08-2013	24-08-2013
Side Quest 1.1: Runic Carpets	200	21-08-2013	27-08-2013
Mission 2: Beyond the Second Dimension	600	26-08-2013	28-08-2013
Side Quest 2.1: Magic Efficiency	300	26-08-2013	30-08-2013
Contest 2.2: Beautiful Runes	400	26-08-2013	06-09-2013
Contest 2.3: 3D Runes	400	26-08-2013	06-09-2013

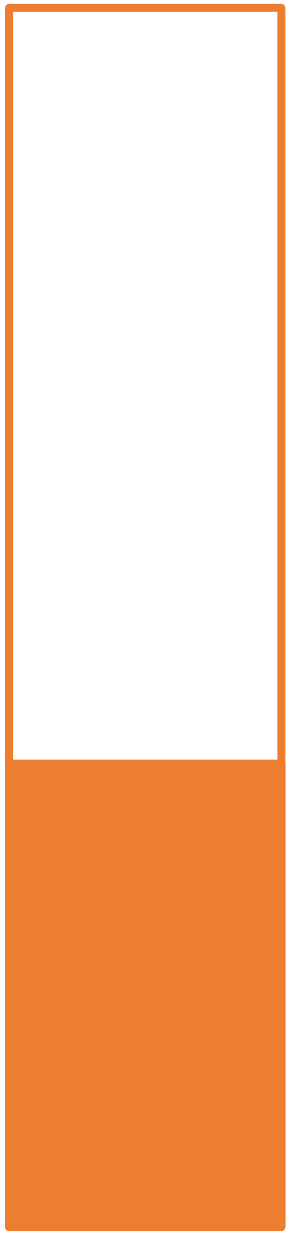
15 Main Missions

+ Side quests & Contests
+ 24-Hour 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

WIN CONTESTS

ATTEND TUTORIALS

COMPLETE SIDEQUESTS

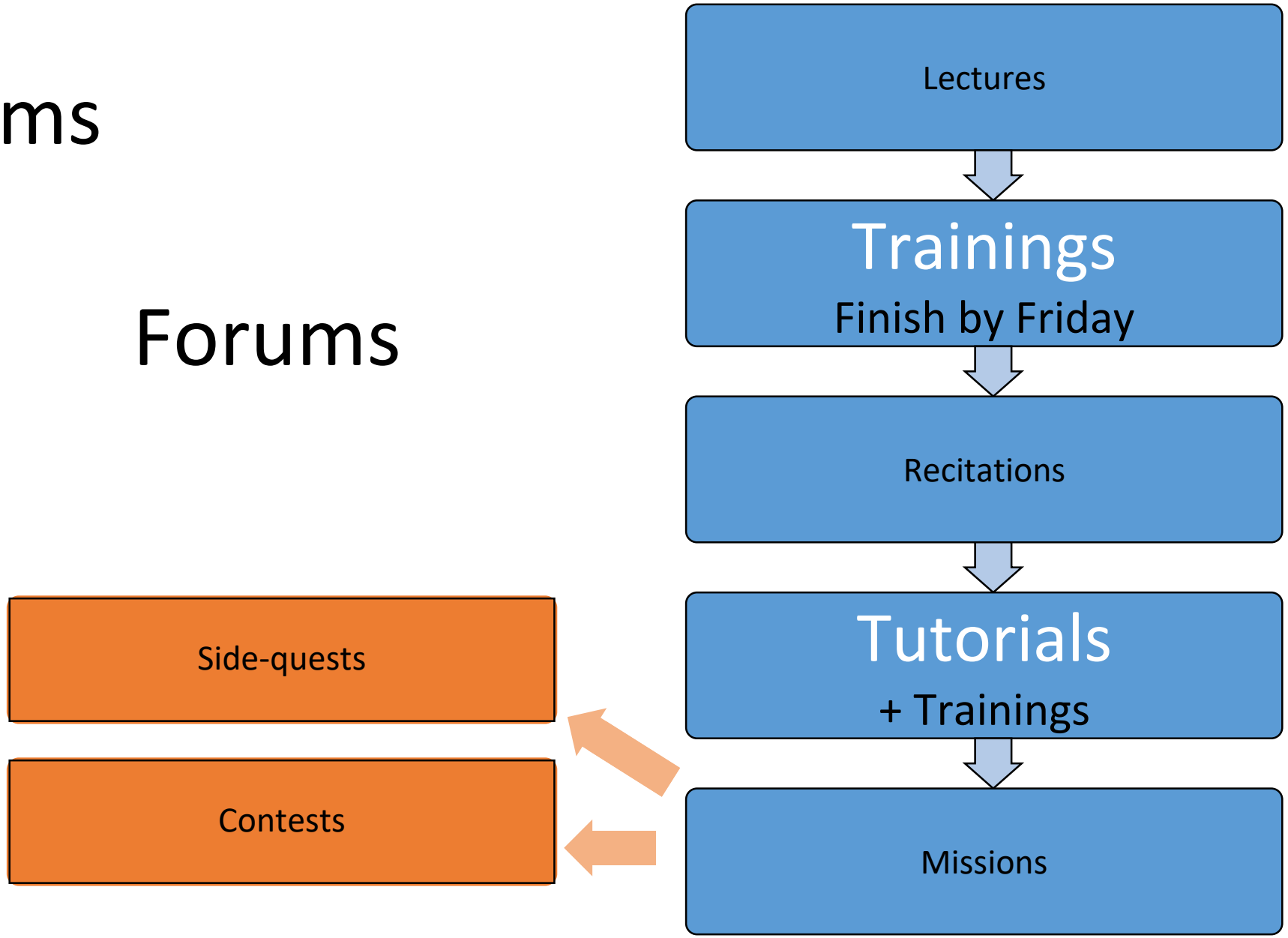
POST ON FORUM
(Reflections)

ENOUGH XP?

LEVEL UP!

Exams

Forums



Assessment Overview

Coursemology	15%
Participation	5%
Mid-term Exam	20%
Practical Exam	20%
Final Exam	40%

Discussion Forum

Do Subscribe

Some Admin Issues!

1. Important Dates

23 Jan, 24 Jan: Chinese New Year

01 Mar: Mid-term Exam (4pm)

07 Apr: Good Friday

15 Apr: Practical Exam
29 Apr: Final Exam (9am)

2. Scheduling problems

- Fill in The Tutorial and Recitation Survey on Coursemology
  (DON'T USE MODREG!)

- Let us know of any clashes with the midterm and the final exam well in advance.

3. Plagiarism Policy

- Zero marks for the affected missions

- Removal of S/U privilege.

Don’t be under the impression
“No one would bother if I copy my friend’s/senior’s code.”

What constitutes as plagiarism?

- Direct copying
- Referencing other’s solution
- Discussion by sharing code

Then what is okay?

- Discussion without sharing code (physically or electronically)

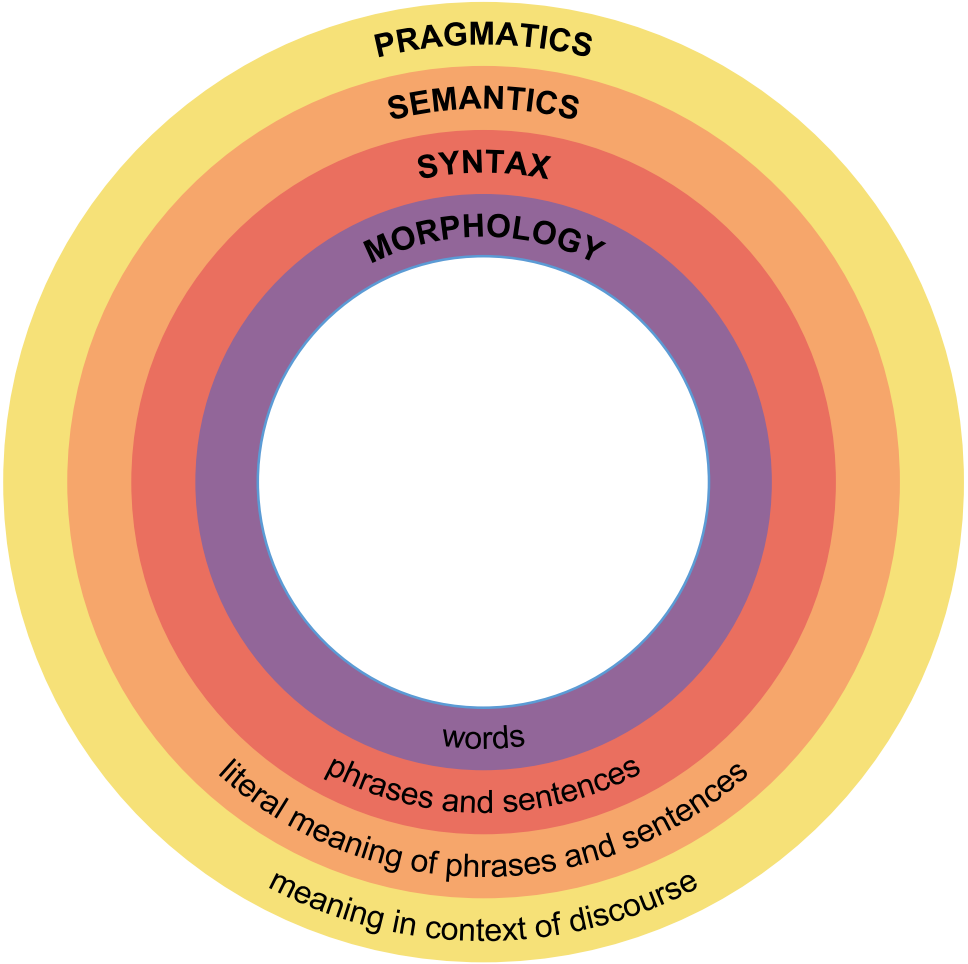
Python

(In few mins)

Elements of a *language*

Programming language

- Data types
- Operators
- Branching
- Iteration
- Means of abstraction



Data Types

Data Type	Definition	Examples	Comments
Integer	A whole number or a number without a fractional part	-1, 0, 1, 1000	Typically have a finite range
Floating point number	A number with a fractional part	-1.5, 0.1, 0.333, 2.56666	They have imprecise representation
Boolean	Two values representing either true or false in a statement in logic.	True, False	Only two permissible values
String	A sequence of symbols	“abc”, “cs1010s”, “10”	Allows us to store human-readable data

Data Types in Python

int	8, 45, 1234
float	2.3, 3.14159
bool	True, False
str	“cs1010s”
	‘cs1010s’
None	

How to check the type?

```
>>> type(123)
<class ‘int’>

>>> type(‘123’)
<class ‘str’>
```

Type Conversion

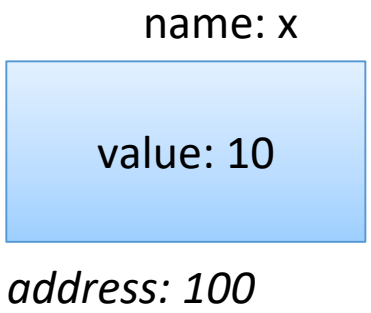
```
>>> str(123)
‘123’

>>> float(‘45.2’)
45.2

>>> int(23.8)
23

>>> int(‘cs1010s’)
ValueError!
```

Variable

- Every variable is characterized by
  - Its name
  - Its value
  - Its address

Its name and Its value → Defined by user

Its address → Assigned by computer



- Every variable can hold only one value at a given instant.
- Old values are overwritten with new values after the assignment.

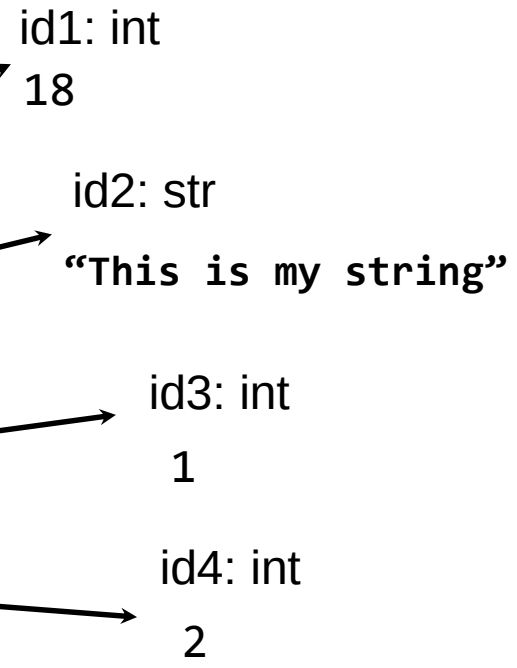
Variable naming convention

- Start with ‘a’-‘z’ or ‘A’-‘Z’ or ‘_’
- Contain only alphanumeric characters or ‘_’
- Case sensitive
- Avoid reserved keywords e.g. `if`
- Python convention: lower case letters separated by ‘_’
  - e.g. `count_change`

Assignment operation

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

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



Arithmetic Operators

+ - * / % // **

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

>>> 2 ** 3
8
```

```
>>> 11 / 3
3.6666666666666665

>>> 11 // 3
3

>>> 11 % 3
2
```

Relational Operators

Truth Values: `True` `False`

`>` `>=` `<` `<=` `==` `!=`

```
>>> 1 <= 10
True

>>> 5 > 15
False

>>> 5 <= 5
True

>>> 2 != 3
True

>>> '1' == 1
False

>>> False == False
True
```

Logical Operators

or and not

```
>>> True or False
True

>>> True and False
False

>>> not False
True
```

OR	True	False
True	True	True
False	True	False

AND	True	False
True	True	False
False	False	False

NOT	
True	False
False	True

Truth Values in Python

- 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`

String Operations

```
>>> 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 Operations

Indexing

```
>>> s = ‘abcd’
>>> s[0]
‘a’
>>> s[2]
‘c’
```

Slicing s[start:stop:step]

non-inclusive

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

Control flow!

If the light is red,

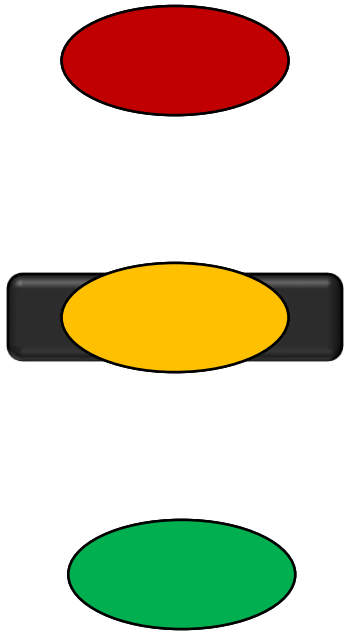
    stop.

Else if it is amber,

    accelerate.

Else,

    proceed with caution.



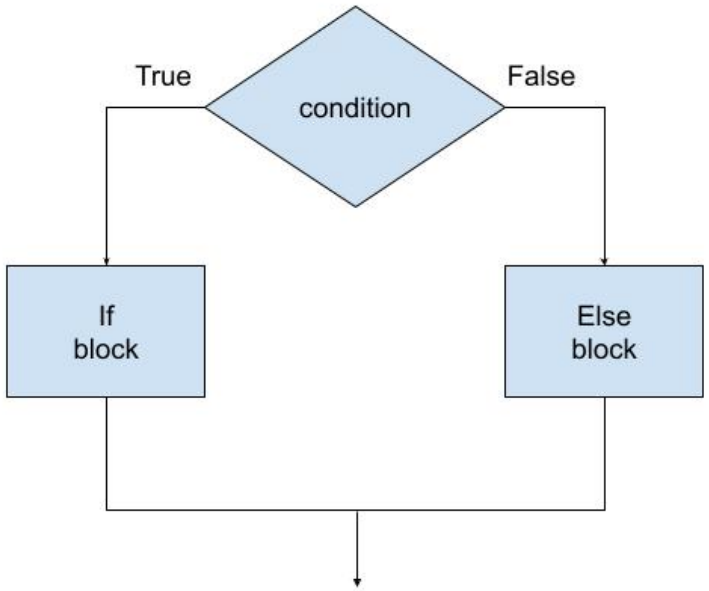
While you are hungry,

eat a spoonful of rice.

Burp.

Conditional Statement

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



expr is a logical expression!

```
a < 2
b >= 3
‘b’ in “hello”
(a < 2) and (b > 4)
...
```

Conditional Statement: Type 1

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

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

'Good'
```

Conditional Statement: Type 2

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

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

'yes'
```

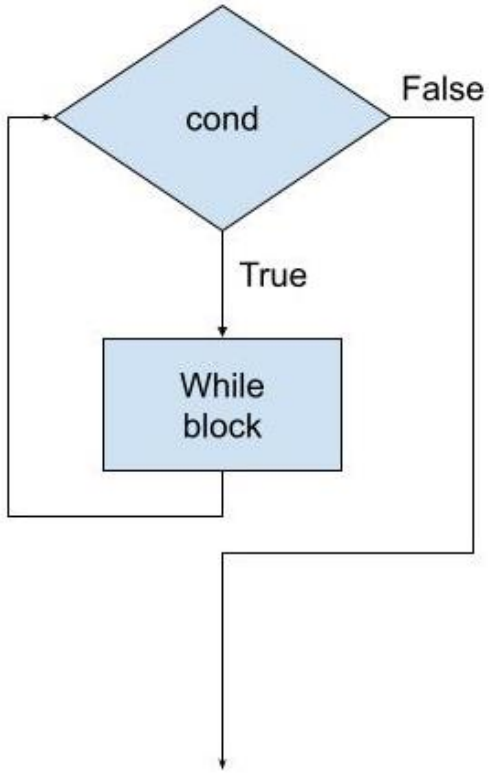
Conditional Statement: Type 3

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

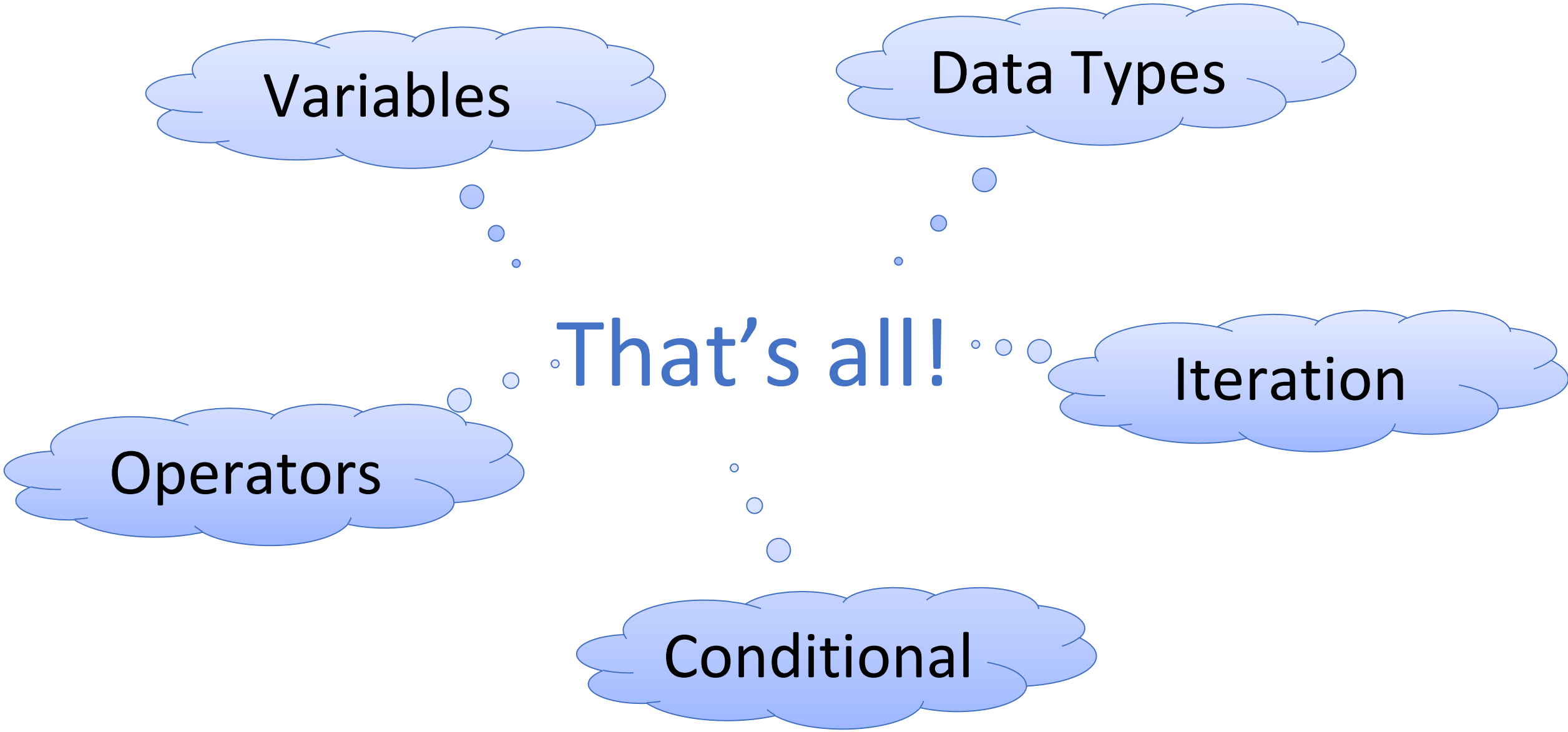
```
>>> a = -3
>>> if a > 0:
        print('yes')
elif a == 0:
        print('no')
else:
        print('huh')

'huh'
```

Iteration: while loop

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

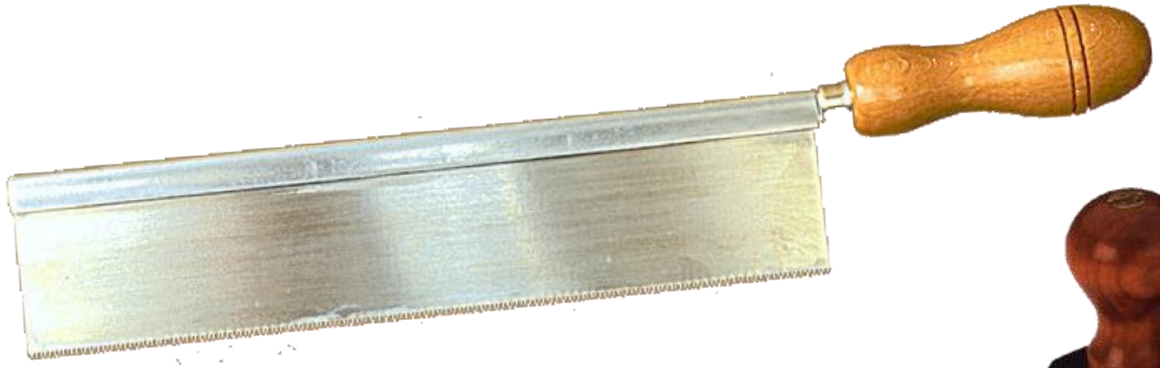



```
>>> a = 0
>>> while a < 5:
        a = a + 1
        print(a)

1
2
3
4
5
```


Variables
Data Types
That’s all!
Iteration
Operators
Conditional

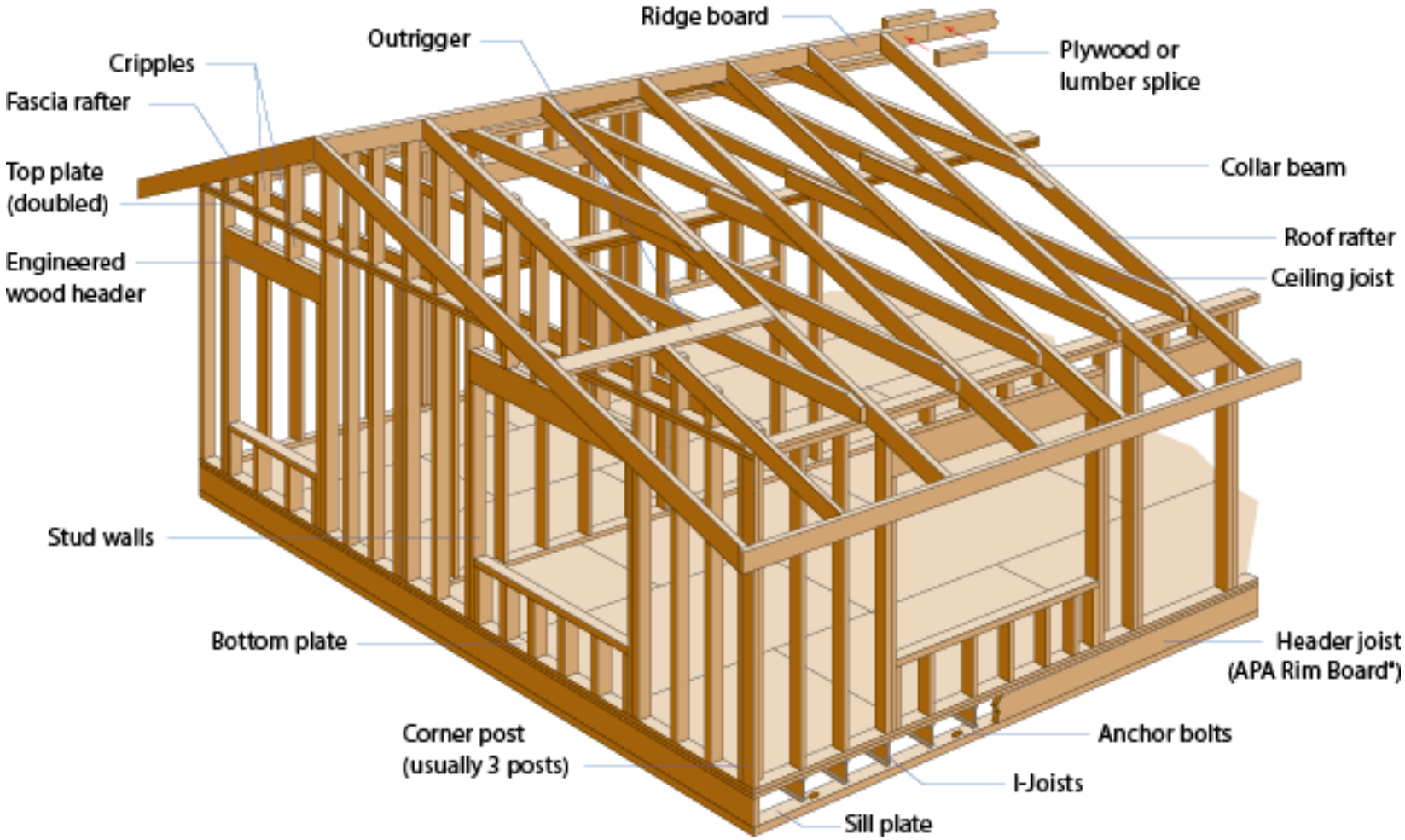
Am I a programmer then?

Are you now a carpenter?



Can you build this chair?

LIGHT-FRAME CONSTRUCTION



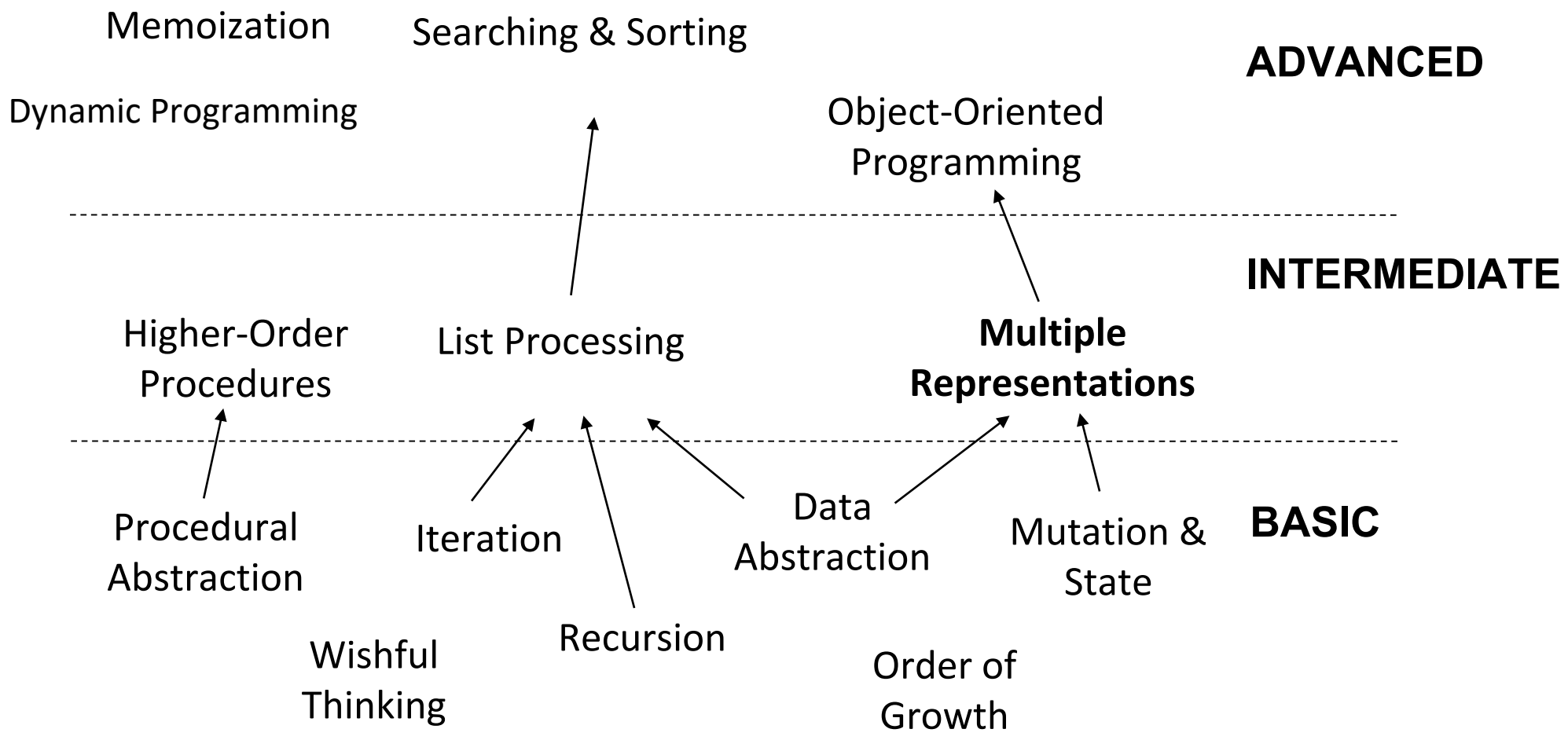
Programming!

- Not just learning the tools
- Not just learning to read code
- Learn to design code

It’s futile to teach languages!

What to learn then? – Computational Thinking!

CS1010S Road Map



Fundamental <u>concepts</u> of computer programming

Practice

Practice

Practice

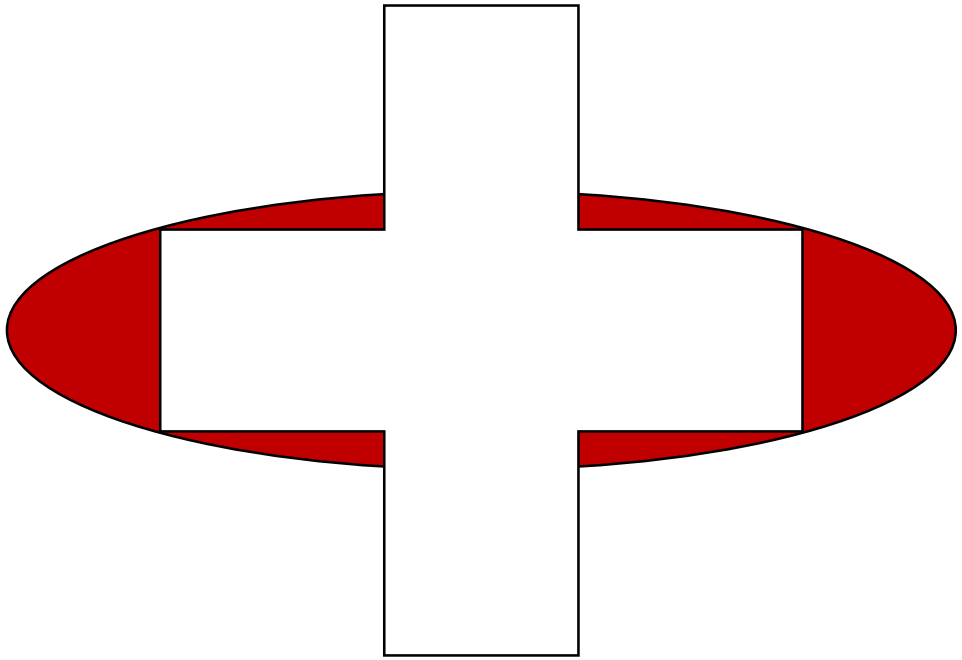
Ways to earn XP

Survey	+100 XP
Training	+100 XP By Friday

Questions?

See you
next week.

Python Clinic



Mission 0

Setting up Python

Recitations	From Week 2!
Tutorials	From Week 3!
Remedial	TBA