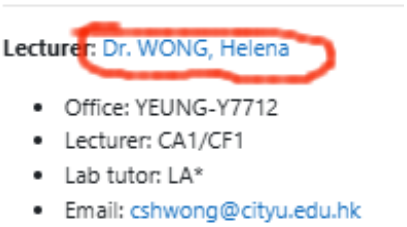
CS1302 - Lecture 2
By Dr Helena WONG
==========
[Canvas => CS1302 = >Home] **scroll down**:
`Contact =>` Dr Helena WONG =>
Supplementary files (e.g. table of contents, Helena's cheat sheets)



==========
Agenda:
* **Warm-up exercises (see below)**
* Lecture 2 - Values and Variables.ipynb
* Lecture 2 - Expressions and Arithmetic.ipynb
==========

Students may try these Warm-up / Break-time exercises:

Q1:
Please create a new notebook for Python, e.g. Untitled.ipynb
Try: type and run **10+6*4**
Try: type and run **2**3**2**

Notes/Tips: You may either
- you may type print(10+6*4) in a cell and run it
- or simply type 10+6*4 in a cell and run it

- But typing two lines like
```
10+6*4
2**3**2
```
in the cell, only the result for 2**3**2 will be shown.

Q2 (a): Try:
```
print('-------------')
print('Hello, World!')
print('-------------')
```

Q2 (b): Single quotes, Double-quotes, Triple-quotes
```
print('-------------')
print("Hello, World!")
print('''-------------''') # triple quotes
print("""-------------""") # triple quotes
```

Q2 (c): The below uses only one print statement. It doesn't work
```
print('-------------
Hello, World!
-------------')
```

Q2 (d): Redo (c) using triple quotes

Q3: Given that ** means "to the power of"
a): 10 + 6 * 4 means?
10+(6*4)? (10+6)*4?
b): 23**2 means?**
2**(3**2)? (2**3)**2?
c): 10 - 6 - 4 means?

10-(6-4)? (10-6)-4?

Notes/Learning:

a) Precedence: * is higher than +

b) ** has Right-assoiciativity or we say *"right-to-left-associativity"*

c) - has Left-associativity or we say *"left-to-right-associativity"*

also for +, etc.

d) Consider 5*6 + 3 ** 7,

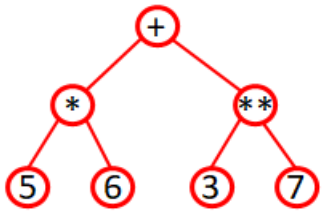
Will python do 5*6 or 3**7 first?

Answer: 5*6 is done first (not 37)**

Student asks: "**But ** has higher priority (precedence)!!**"

Explanation:

The evaluate of an expression is based on its structure (a binary tree)



precedence (priority) doesn't mean run FIRST or run NEXT

precedence (priority) is for grouping (how the expression structure is parsed)

In **5*6 + 3 ** 7,**

** has higher precedence than +.

Therefore the result of 3**7 becomes an operand of +.

That is:

i) ** has highest priority. So, ** first takes 3 and 7 as its operands.

ii) * and + are left. Because * has higher priority,* can then take 5 and 6.

iii) Finally + takes its operands: (5*6) and (3**7)

Well, for (5*6) + (3 ** 7), you and me will do 5*6 first, agree? Same concept for Python ^_^

Conclusion: There are actually 2 steps:

Step 1: understand 5*6 + 3**7 as (5*6) + (3 ** 7)

Step 2: based on (5*6) + (3 ** 7), calculate the left side (5*6) to get 30, then calculate the right side (3**7) to get 2187, then add 30 and 2187 to get 2217.

Q4: What are the following:

10**0.5, (10**0.5)**2

10==10, 10==10.0, 10==10+0.01, 10==(10**0.5)**2

import math

math.isclose(10, 10.0001), math.isclose(10, (10**0.5)**2)

Think:

When I borrow $2 from you. Then I return $1.1. The record says I still own you $0.89999.

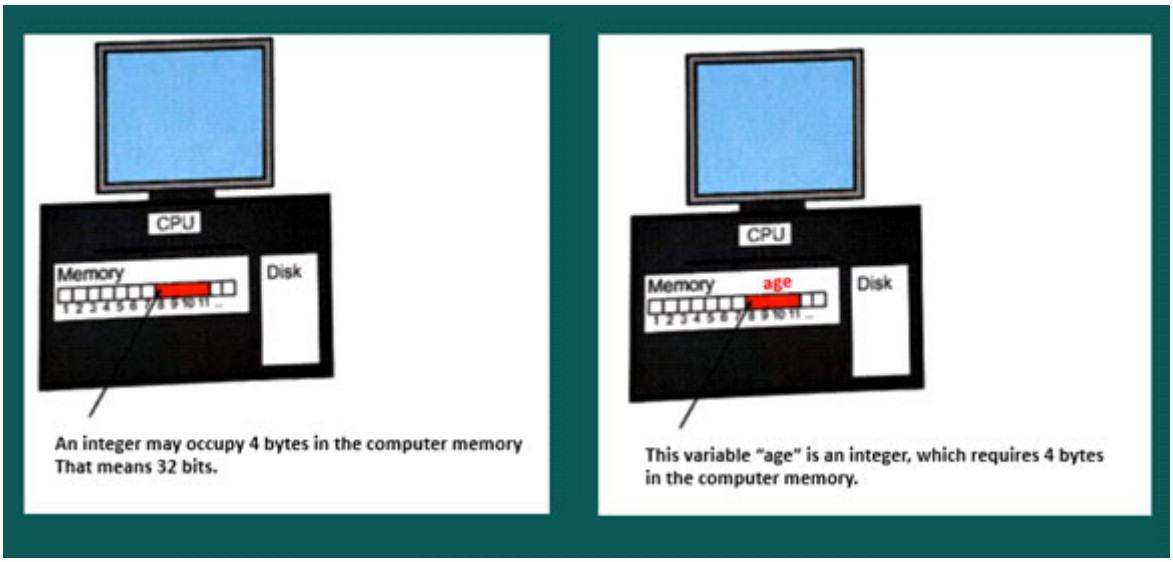
When I return you $0.9, the record is not made zero.

What if similar is done in banking system?

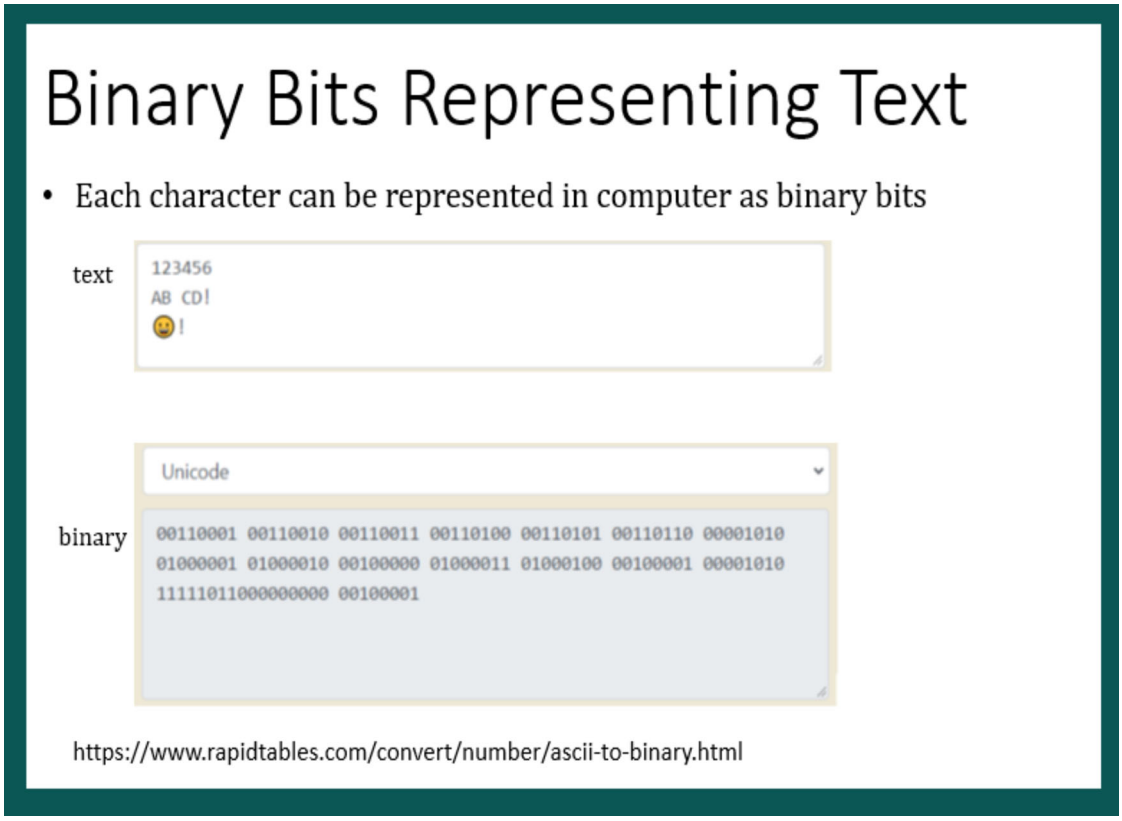
==========

Values and Variables

==========



Strings



https://www.rapidtables.com/convert/number/ascii-to-binary.html
123456
AB CD!
😀!

* google: unicode 1f600
https://www.compart.com/en/unicode/U+1F600

```
print("#")
print("$")
print("%")
# print(""")
```

Escape sequence

print("\"") # "\..." : "escape sequence"

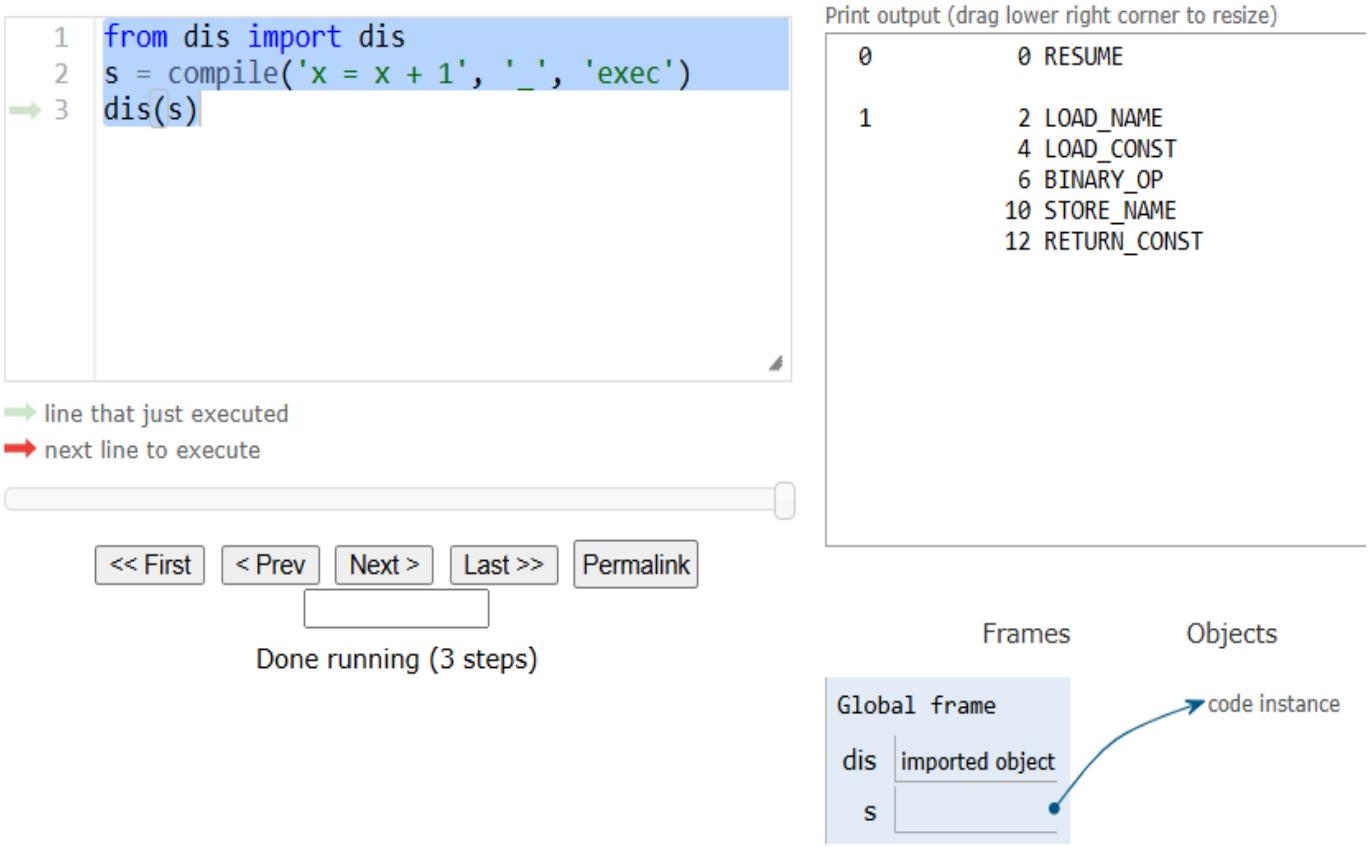
\" : double-quote
\U... : a unicode
\N
\n
\t

```
print('\U0001f600', 'I am hungry \U0001f601')

print("\N{grinning face}:",
    "\tI'm a string.",
     sep="\n")  # separator: change from space (default) to newline
```

dis(compile(...))

```
from dis import dis
s = compile('x = x + 1', '_', 'exec')
dis(s)
```



---

3 kinds of program errors:

(1) Syntax error (cannot convert to runnable code to even start running)
(2) Runtime error (code stopped during running)
(3) Logical error (can finish, but wrong output)

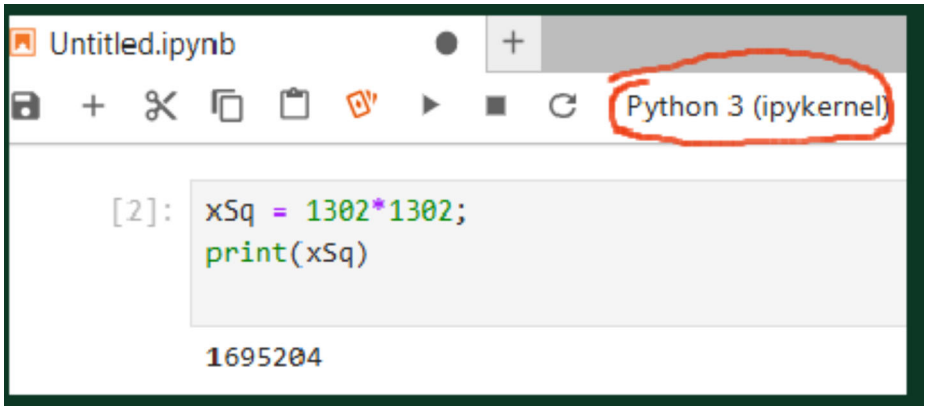
Python is a strongly-and-dynamically-typed language:
(c++ strongly-and-statically-typed language)

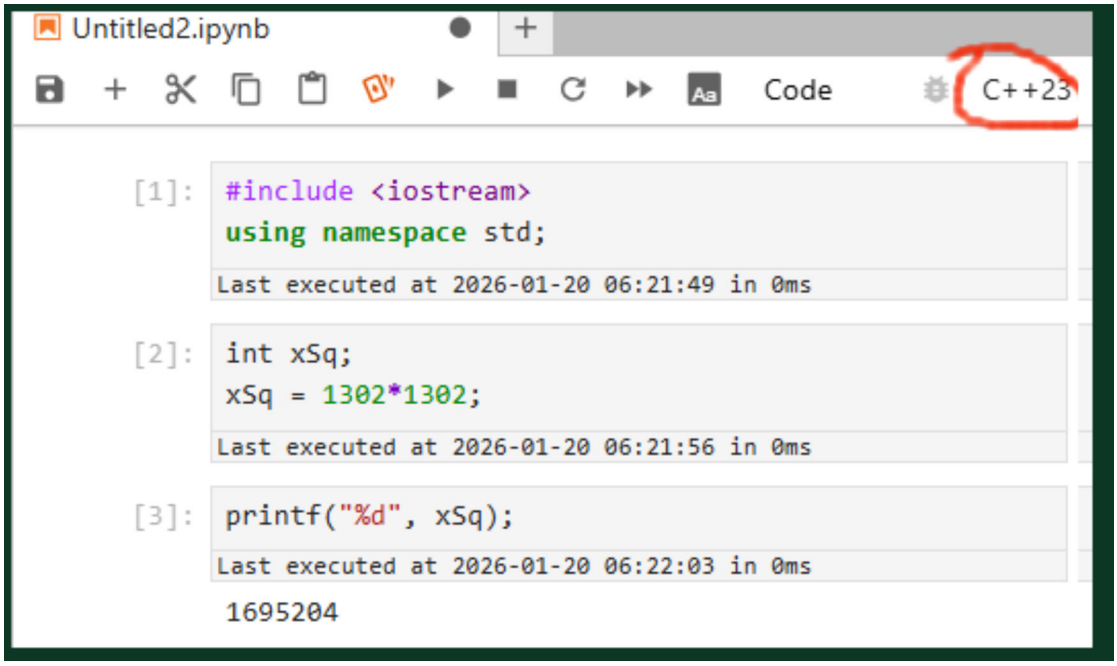
======= Python =======

xSq = 1302*1302;
print(xSq)
======== C++ =========
#include <iostream>
using namespace std;

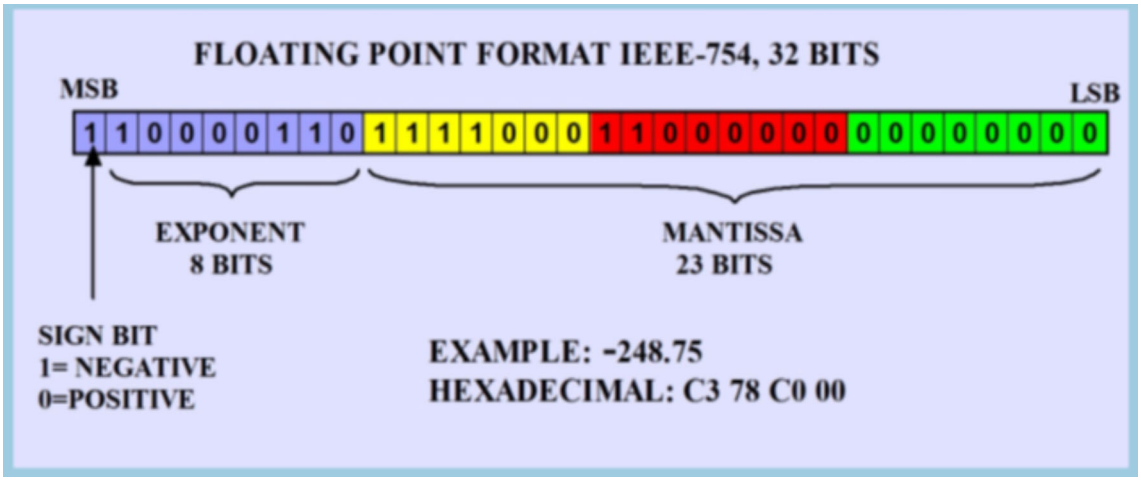
int xSq;
xSq = 1302*1302;

printf("%d", xSq);
=====================





==========
Expressions and Arithmetics
==========



https://www.puntoflotante.net/FLOATING-POINT-FORMAT-IEEE-754.htm

Internal representation of Floating point numbers (able to handle very small or very large):

The number is translated to

one point something **times** two **to the power of** something (in binary)

e.g. -248.75 => -1.xxxx * 2 ^{yyy}

The exponent (8 bits) stores yyy+127 (so that yyy can be -127 to +127)
The mantissa (23 bits) stores xxxx
The sign bit is 1 or 0 for -ve or +ve

"I have $1234891234.234189"

=> (If only the 5 most significant digits are stored):

$1234800000.000000

What is 0.7_{10} in binary?
https://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html