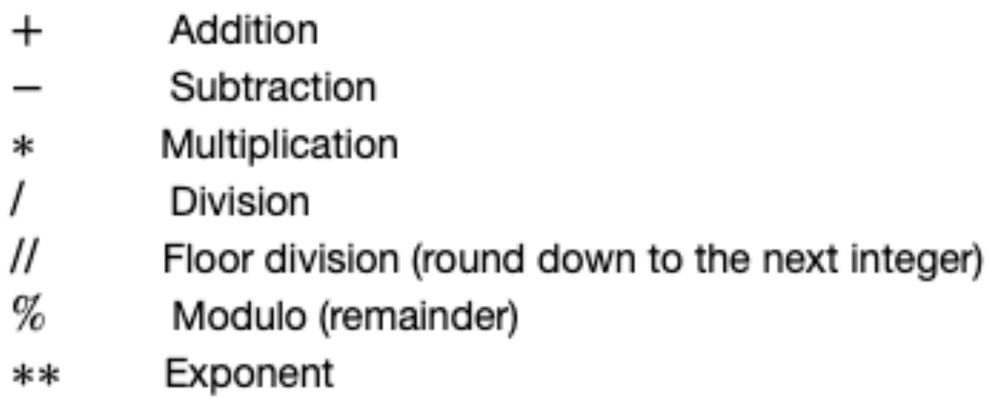
Introduction to Computer Programming
Lecture 11.1:

Review :
Operators and Data Structures

Hemma Philamore

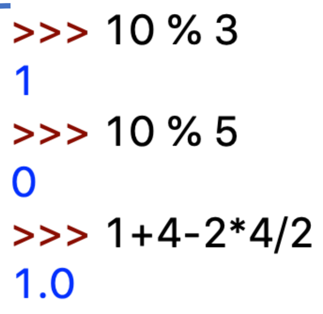
Department of Engineering Mathematics

Mathematical operators



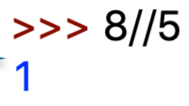
+	Addition
−	Subtraction
*	Multiplication
/	Division
//	Floor division (round down to the next integer)
%	Modulo (remainder)
**	Exponent

% modulo



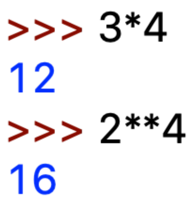
```
>>> 10 % 3
1
>>> 10 % 5
0
>>> 1+4-2*4/2
1.0
```

// floor division



```
>>> 8//5
1
```

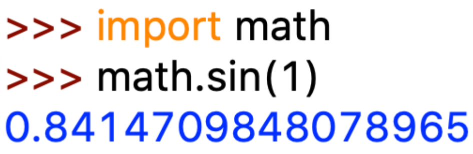
multiplication



```
>>> 3*4
12
>>> 2**4
16
```

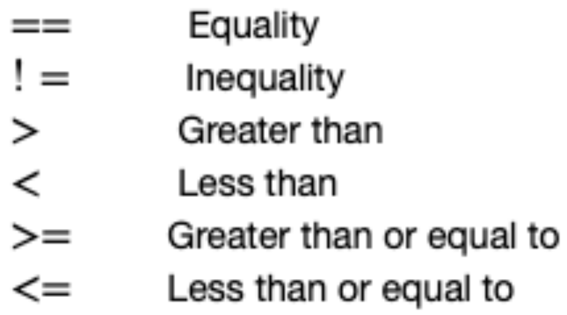
exponent

additional functionality



```
>>> import math
>>> math.sin(1)
0.8414709848078965
```

Boolean operators



==	Equality
! =	Inequality
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

```
>>> 1 > 2
False
>>> 3 <= 5
True
>>> 1 == 1
True
>>> 1 != 2
True
```

Logical operators

Compare the outcome of two Boolean operations

and : both are true
or : either are true

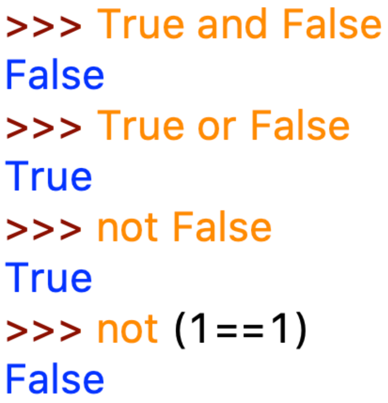
not: negates the outcome of a conditional

```
>>> (1 > 2 and 3 < 4)
False

>>> (1 > 2 or 3 < 4)
True
```

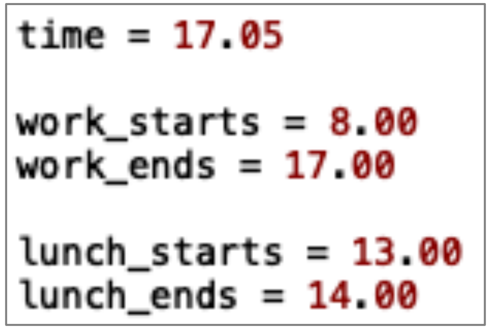
```
>>> not(1 > 2 or 3 < 4)
False

>>> not(1 > 2) and 3 < 4
True
```



```
>>> True and False
False
>>> True or False
True
>>> not False
True
>>> not (1==1)
False
```

Operators are combined to produce compound statements



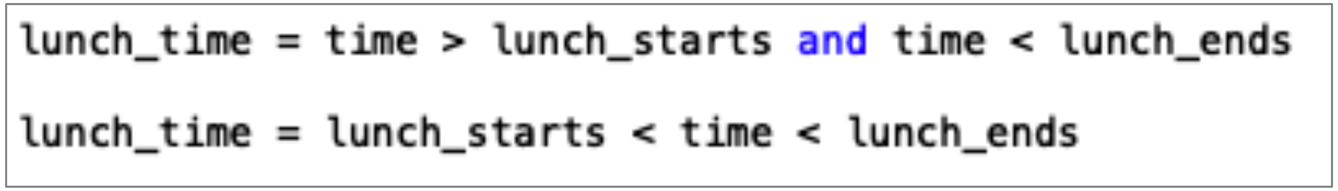
```
time = 17.05

work_starts = 8.00
work_ends = 17.00

lunch_starts = 13.00
lunch_ends = 14.00
```

Change the value of **time** to change the value of **lunch_time** and **work_time**

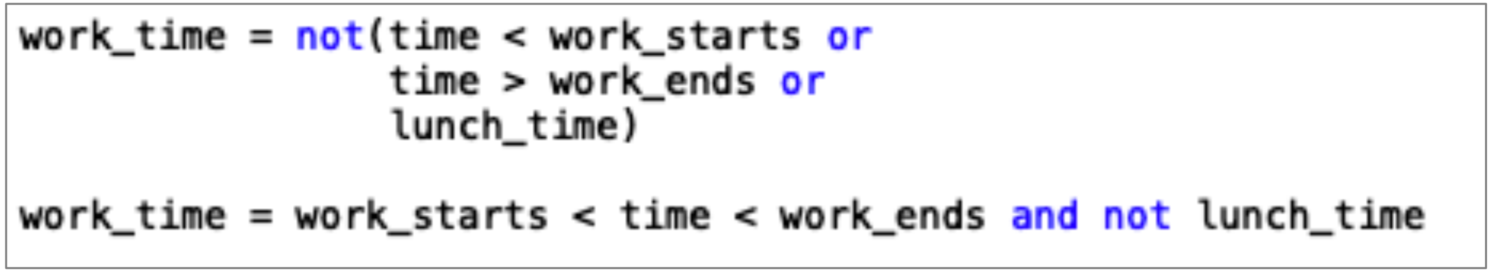
Lunchtime is after lunch begins and before lunch ends



```
lunch_time = time > lunch_starts and time < lunch_ends

lunch_time = lunch_starts < time < lunch_ends
```

Work time is after works begins, before work ends, and not at lunchtime



```
work_time = not(time < work_starts or
                time > work_ends or
                lunch_time)

work_time = work_starts < time < work_ends and not lunch_time
```

Q.11.1.A

Express these statements using mathematical, logic and Boolean operators.

a) An increasing population is when the number of new emigrants and the number of deaths is less than the number of new immigrants and the number of births.

b) Draw two counters of the **same colour** from a bag of blue and green counters on the **first turn** and then draw two counters of **different colours** on the **second turn** to win the game.

Data Structures

Lists

my_list = [1, ‘five’, 3.2]

An *ordered* collection of items

Mutable : can be altered
(items added, removed, changed)

Tuples

my_tuple = (cos(θ) , sin(θ), 1)

Holds multiple items together

Immutable : cannot be modified

Sets

my_set = {2, 2, 1, 2, 5}

Unordered collections of simple objects.

Used when we are concerned with the existence of an object or membership of a set (rather than exact objects, position in the data structure, or number of occurrences)

Dictionaries

my_dict = { ‘Engineering Maths’ : 75,
‘Digital Heath’ : [25, 4]
}

Address-book style collection of keys (name) and value (details).

Keys: only immutable objects
Values: mutable/immutable objects