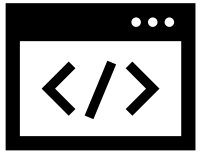
Imperial College London



BIOE50010 – Programming 2

Computer Lab 7

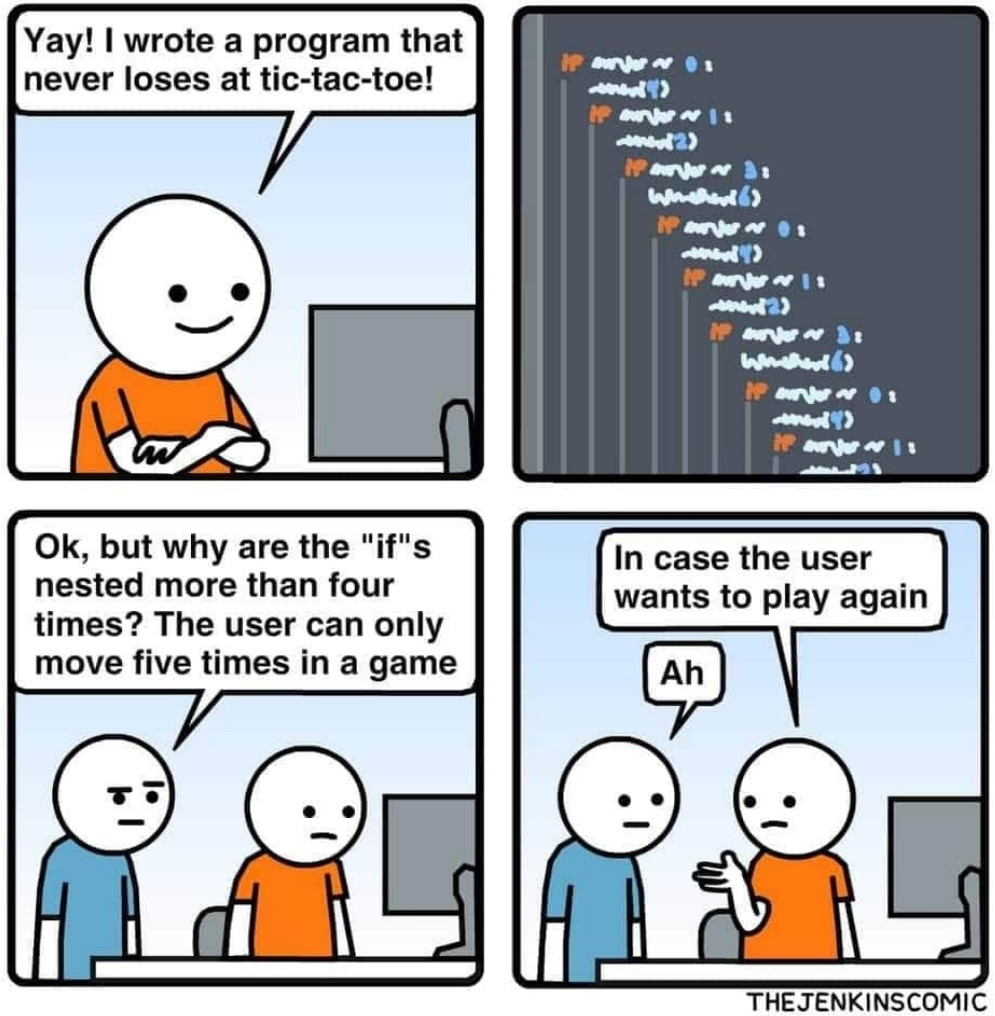
Binghuan Li | *Department of Chemical Engineering*

binghuan.li19@imperial.ac.uk

November 20, 2023

Imperial College London

Meme of the week ☺



Source: https://twitter.com/codeforcauseIn/status/1470744970239700993

Imperial College London

Last Week - Four Pillars of OOP

1) ***Abstraction*** focuses on the essential characteristics of an object relative to the perspective of the viewer.

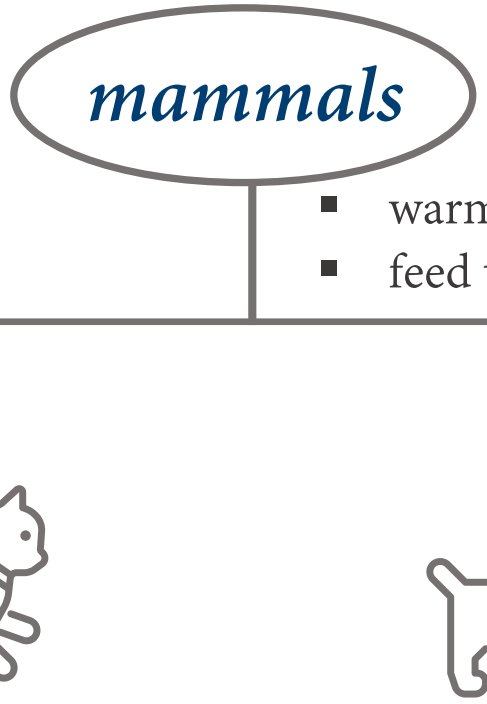
2) ***Encapsulation*** hides the details of the implementation of an object.

3) ***Inheritance*** allows for a derived object type to inherit features from another object type.

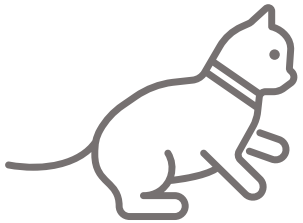
4) ***Polymorphism*** allows for overriding any inherited method by creating your own method within its own class.

Source: https://www.reddit.com/r/ProgrammerHumor/comments/gu4k3y/oops/

Inheritance

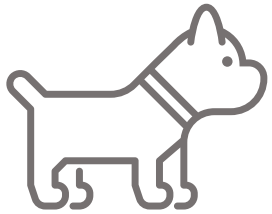


- warm-blooded
- feed their young with milk



also…

- can't taste sweetness
- powerful paws



also…

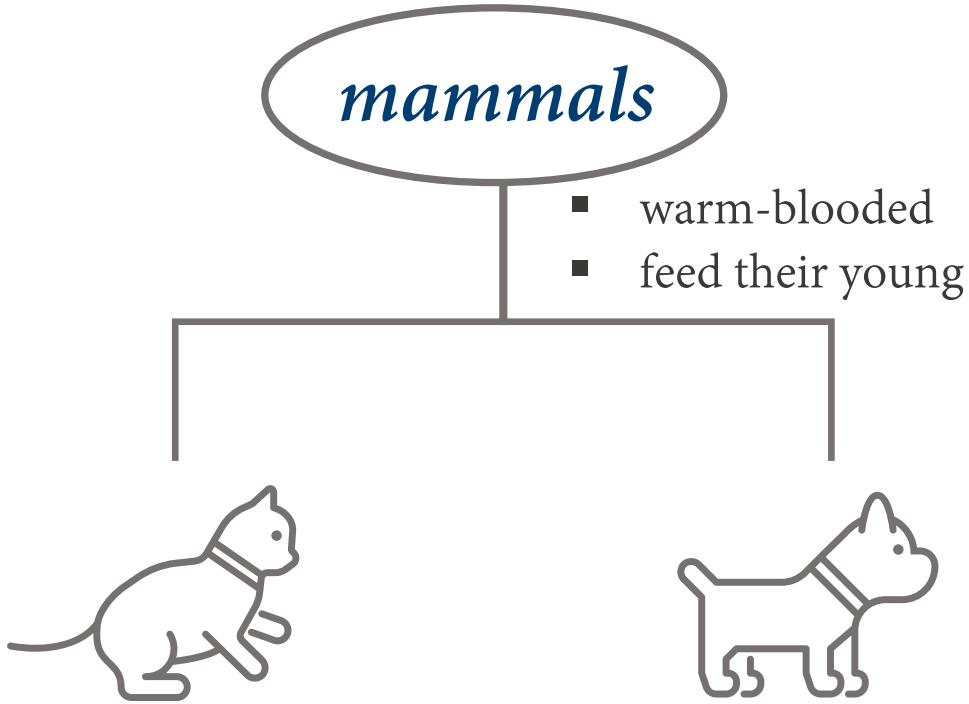
- good sense of smell
- can detect high-pitched noises

Common characteristics in a (parent) category

keeping **common** characteristics while deriving new **unique** characteristics.

Unique characteristics in a sub-categories

Inheritance



- warm-blooded
- feed their young with milk

also…

- can't taste sweetness
- powerful paws

also…

- good sense of smell
- can detect high-pitched noises

Example

```
class Mammals:
    def __init__(self):
        pass;
    # common features go here…

class Dogs(Mammals):
    def __init__(self):
        pass;
    # unique features go here…
```

Imperial College London

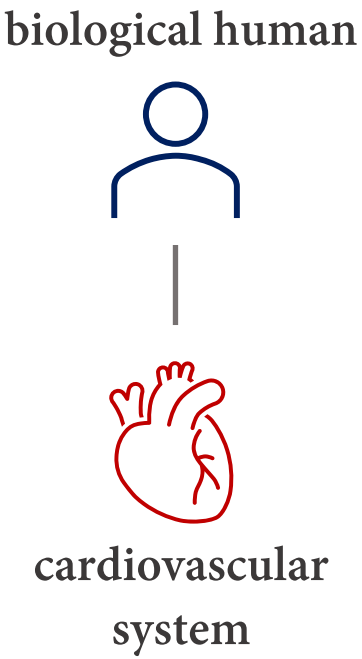
Other Relationships Between Classes

- Inheritance depicts an "***is-a***" relationship between the super class and subclass.
  - `Dogs` is a subclass of `Mammals`
- There also exists a "***has-a***" relationship between two classes:
  - A biological human has the cardiovascular system
  - The Dept. of Bioengineering has 57 academics
  - A car has an engine
- Such relationships are described by **composition** and **aggregation** in OOP.

Imperial College London

Advanced Topic 1 - **Composition**

- ***Composition*** is a **strong** form of association where a class contains objects of another class as part of its internal structure.



"The ***cardiovascular system*** is a vital part of the human body."

Example

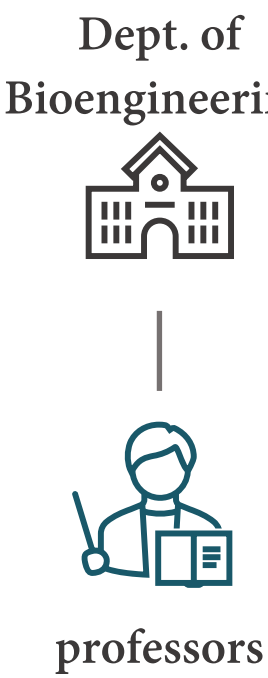
```
class Cardiovascular:
     def __init__(self):
          pass;
     # implementation of cardiovascular system

class Human:
     def __init__(self):
          self.cardiovascular = Cardiovascular();
     # a human body composes of a few objects
```

Imperial College London

Advanced Topic 2 – Aggregation

- ***Aggregation*** is a **weak** form of association where a class contains objects of another class as part of its attribute, but doesn't manage the lifecycle of the objects.

"A ***department*** may have an aggregation relationship with a ***professor***, but a professor can also exist independently."



Example

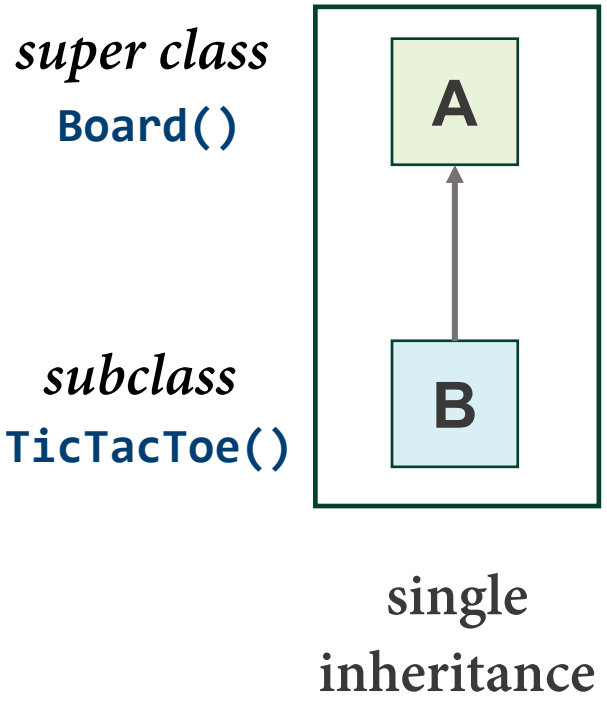
```
class Department:
     def __init__(self):
          self.professors = []

class Professor:
     def __init__(self, name):
          self.name = name

bioeng = Department()
professor1 = Professor("Prof. Weinberg")
bioeng.professors.append(professor1)
```

Note the difference here from the composition!

Imperial College London

Your task today (1/)

- Code the *Tic Tac Toe* game in the object-oriented fashion.
  - **`Board()`** class: a ***base*** (super) class that are generalised to be suitable to any board game.
  - **`TicTacToe()`** class: a ***derived*** (sub) class for the game TTT that inherits the attributes and methods defined in the base class **`Board()`**.
  - **`main()`** function used to initiate the game is given in the sample code.

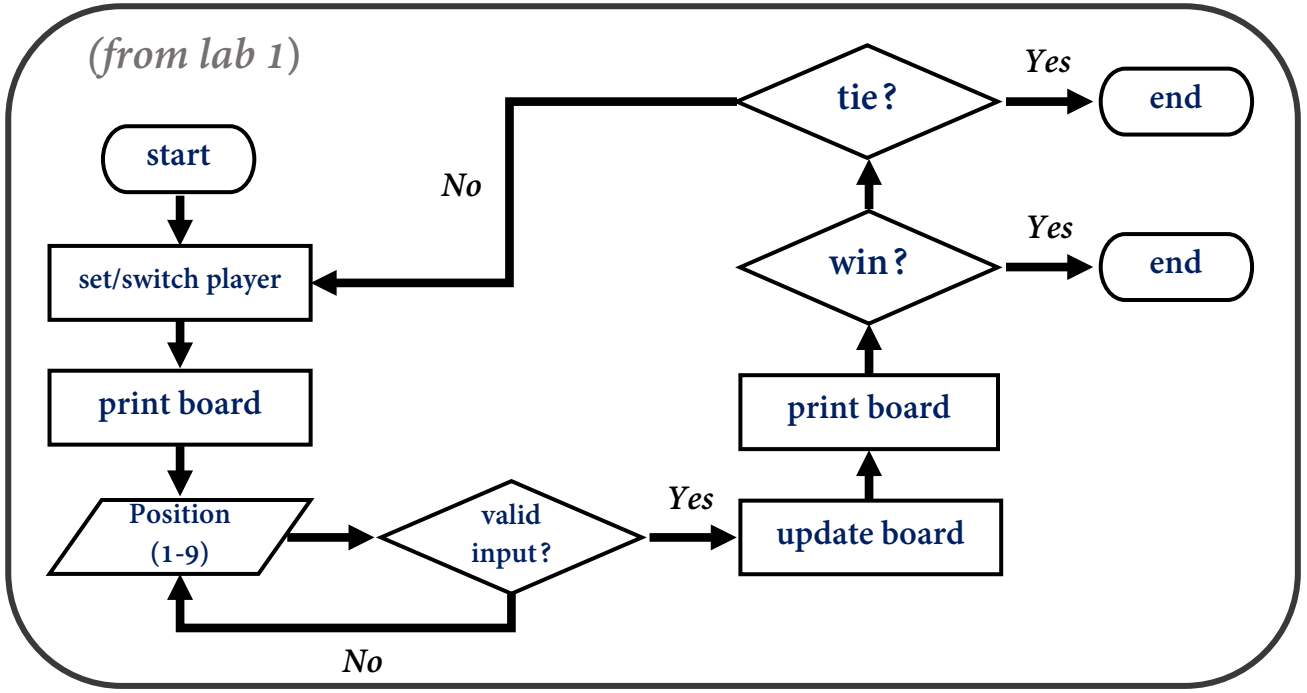


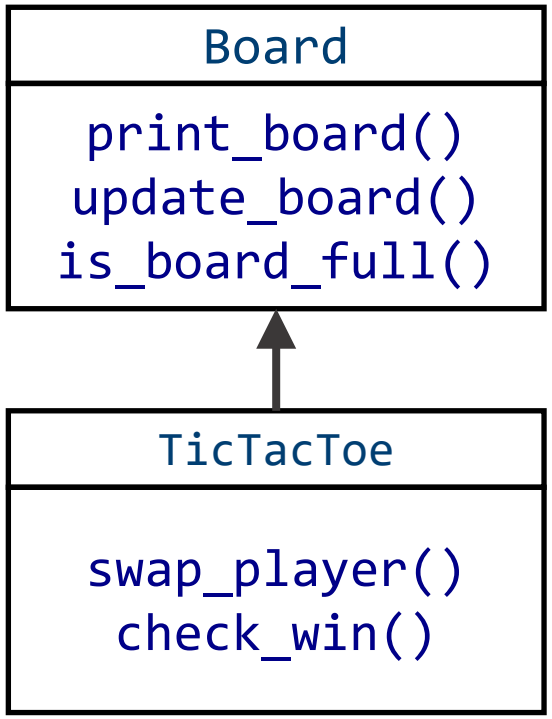
- Read the sample code and console output carefully before you start.
- We only consider ***inheritance*** today!

Your task today (2/)

Q: For each class, what methods do I need to define?

A*:** what features (procedures) are ***common for all board games? What features (procedures) are ***unique*** for Tic Tac Toe only? **Very task-specific!**





Imperial College London

Questions?

That's it for now.

You may now proceed to the Lab 7 exercise.