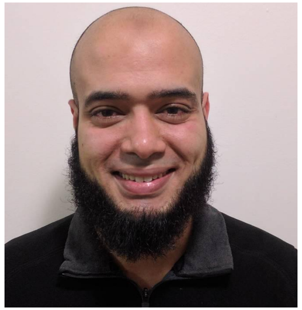
Python Programming
Inheritance in Practice

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Inheritance in practice

- In past = major technique for reusability and extensions
- Now = A lot of careful before using it (E.g. as in homeworks)
  - ***Prefer composition over inheritance***
  - Avoid as **much** as possible inheritance. Use inheritance if you have **strong** justifications
    - It is really **is-a** relationship.
    - Parent class is superclass for all subclasses.
      - Think deeper about future changes
      - But future is really hard to predict :(
    - You don’t do it just to do some **code reuse**

Multiple Inheritance in practice

- Avoid it. Avoid it. Avoid it unless it is really a good one
    - With minor mistakes: you may end up with e.g. uninitialized base classes or errors for missing parameters. It is also a source of confusion.
    - Prepare strong justification for your team
    - Make the inheritance hierarchy a tree style
- One clear issue: if we have arguments from a class to another, then?!!
    - This is a big issue
    - One popular workaround: Cooperative Multiple inheritance
        - Core concept: use **kwargs in args + all calls super()
    - If you are using someone multiple inheritance, make sure to understand it / above link
    - Future readings: link link link link link

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."