Python Programming

Data-Hiding

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)

Name mangling

- In the last video we learned about this process
- Some attributes/methods are not directly accessible from outside
  - The are meant to be used **internally** (inside the class)
  - But python doesn't prevent you from really forcing an access
- So it is kind of **data-hiding**, but not so strict (*weakly-private*)
- Useful to avoid **name collisions** in inheritance hierarchy
- It is educative to introduce another perspective from languages like C++/Java

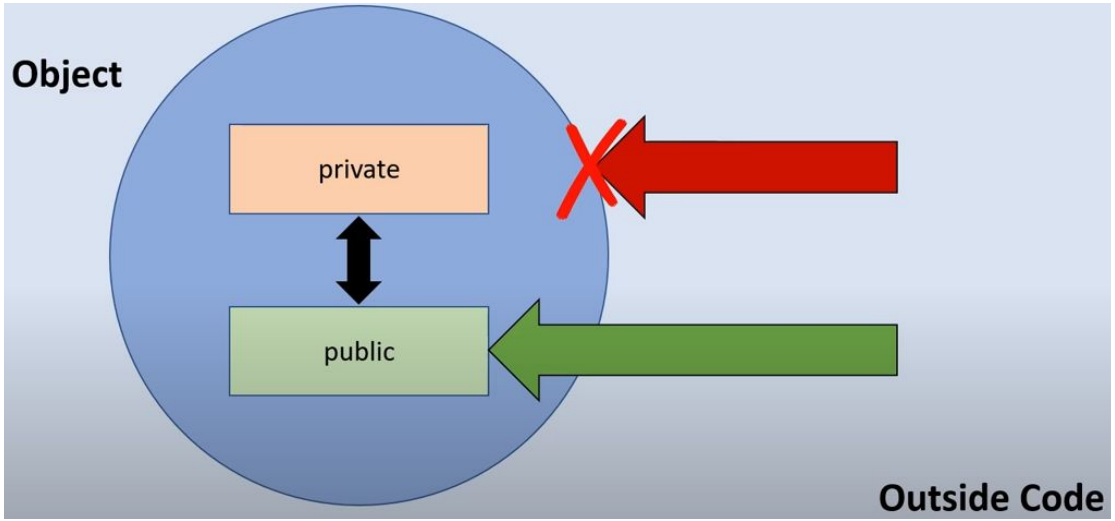
C++ Class: Private vs Public

- Similar to python, but is divide to 2 parts
- **Public** section:
  - **Outsider** can see/access its attributes/methods
- **Private** section:
  - **Outsider can't** access
  - Insiders only can see/access

```
class Quote {
private:
    int internal;

    string GetQuote() {
        return "The way to get started is to "
               "quit talking and begin doing";
    }

public:
    int external;

    Quote() {
        internal = 3, external = 7;
    }

    void print() {
        cout<<GetQuote()<<"\n";
    }
};

int main() {
    Quote q;

    cout<<q.external<<"\n";
    q.print();

    //q.internal;       // can't access
    //q.GetQuote();     // can't call
    return 0;
}
```

C++ Data Hiding Concept

- The private section **hides** some data members & member function **from user**
  - Users (outside code) are either other classes in same project or client using final project
  - A **good design**: reveal as little as possible of the data members & functions



Img Src

C++ Data Hiding: WHY?

- To **prevent corruption of data** by other entities (outside code).
  - Such changes might be unintended or intended
- Protect object's data ⇒ protects object integrity
  - Imagine you have a computer desktop (mobile/car) that has a problem
    - You figured out there is problem in xxx (e.g. adapter of laptop)
    - You bought new cheap yyy similar to xxx but not right model
    - EIther it won't work or work temporarily then fails soon (e.g. voltage problem)
    - Integrity fails as whole system components are not proper now
- Data hiding also reduces system complexity
- Better code readability (less complex code is viewed).

Back to Python

- Python takes the opposite direction.
  - By default, we leave things public/accessible. The last resort is to restrict
  - We assume responsibility
  - Is it language limitation as not a compiled language? Or culture?
- Cases:
  - In doubt: leave it public / no mangling
  - Share intention of *'please don't touch'*? Just use single underscore (**_name**)
  - Some disaster might happen if was abused? Use __
- Python vs C++: which is better approach? Controversial
- Future reading

Coming from C++/Java

- All programming languages share a lot of things
- But still there are different philosophies/cultures
- Your mind is tuned to C++/java, you need time to do mentality shift
- Don’t do things as you used to do in them.
  - Take a step back.
  - Think/Search how to make things Pythonic
- Be open to different philosophies/cultures. Give a series trial.
- Moving later from Python to C++/Java
  - You will write much more code!
  - Different mind set
  - Much more language constructs such in modern C++ (seriesly a complex language)

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."