Object Oriented Programming

### WHY OOP?

* OOP is most advantageous when a program needs to be structured around real-world entities that have both data and behavior.
* Key strengths of OOP:
  + Code organization and reusability
    - OOP organizes code into self-contained objects, which makes it easier to manage large, complex applications. The principle of inheritance allows for the reuse of code by defining new classes based on existing ones, reducing redundancy and saving development time.
  + Modularity and maintainability
    - By breaking down large systems into smaller, independent objects, OOP simplifies code maintenance. A change or bug fix in one object is less likely to affect the rest of the program, which isolates issues and simplifies troubleshooting.
  + Enhanced data security
    - Encapsulation, a core OOP principle, bundles data and methods into a single unit (an object) and restricts direct access to the data. This protects an object's internal state from unintended external modification, which improves security and integrity.
  + Flexibility and scalability
    - Polymorphism allows objects of different classes to be treated as instances of a common superclass, enabling flexible and extensible systems. As a project grows or new requirements emerge, OOP makes it easier to add new features or modify existing ones without altering the core codebase.