Abstraction

Abstraction is an object-oriented programming concept that allows a programmer to hide the implementation details of a class or method and only expose the essential functionality to the user.

Python provides abstraction mainly through abstract classes and abstract methods, which are defined in the abc module (ABC stands for Abstract Base Class).

Abstract Class:

- A class that cannot be instantiated directly.
- Can contain abstract methods (methods without body) and regular methods (with implementation).

Abstract Method:

- A method declared but not implemented in the abstract class.
- Must be implemented in a subclass.

Concrete Method:

- A concrete method is a regular method in a class that has a full implementation (it contains actual code).
- Unlike abstract methods, which only have a declaration (pass) and must be implemented in subclasses, concrete methods can be used directly.

```python
from abc import ABC, abstractmethod

# Abstract Class
class Vehicle(ABC):
    @abstractmethod
    def start(self):    # Abstract Method
        pass
    @abstractmethod
    def stop(self):         # Abstract Method
        pass
class Car(Vehicle):
    def start(self):
        print("Car starts with a key.")

    def stop(self):
        print("Car stops with a brake.")

class Bike(Vehicle):
    def start(self):
        print("Bike starts with a self-start button.")

    def stop(self):
        print("Bike stops with disc brakes.")

v1 = Car()
v1.start()
v1.stop()

v2 = Bike()
v2.start()
v2.stop()

# Output
Car starts with a key.
Car stops with a brake.
Bike starts with a self-start button.
Bike stops with disc brakes.
```