Data Abstraction

Pankaj Chouhan

www.codeswithpankaj.com

Data abstraction is a concept in object-oriented programming that hides unnecessary details from the user and only shows the essential features of an object. It helps in reducing complexity and increasing code readability.

How Does Abstraction Work ?

- In Python, abstraction is achieved using abstract classes and abstract methods.
- An abstract class is a class that cannot be instantiated (you cannot create an object of it).
- It contains abstract methods (methods without implementation) that must be implemented in the child class.

Think of data abstraction like a TV remote control. You just need to know which buttons to press, but you don't need to know how it works inside!

Example using a Mobile Phone

Think of it this way :

- When you use your real mobile phone, you just press the power button
- You don't need to know how the battery works inside
- You just need to know how to check battery level

This is exactly what abstraction does :

1. Hides complicated stuff inside (using __)
2. Gives you simple methods to use (like switch_on())
3. Protects the data from accidental changes
4. Makes the code easier to use

READ MORE

Pankaj Chouhan

www.codeswithpankaj.com

Example

```
class MobilePhone:
    def __init__(self):
        self.__battery_level = 100
        self.__is_on = False

    def switch_on(self):
        self.__is_on = True
        print("Phone is switched ON")

    def switch_off(self):
        self.__is_on = False
        print("Phone is switched OFF")

    def check_battery(self):
        return f"Battery level: {self.__battery_level}%"

# Using the phone
my_phone = MobilePhone()
my_phone.switch_on()
# Output: Phone is switched ON
print(my_phone.check_battery())
# Output: Battery level: 100%
```