Amr Bahaa eldin Mohamed Ahmed Ail

4200491

Abstraction in Python

Abstraction in python is defined as a process of handling complexity by hiding unnecessary information from the user. This is one of the core concepts of object-oriented programming (OOP) languages. That enables the user to implement even more complex logic on top of the provided abstraction without understanding or even thinking about all the hidden background/back-end complexity.

That's a very generic core topic not only limited to object-oriented programming. You can observe it everywhere in the real world or in our surroundings

Abstract Classes and Methods in Python

To declare an Abstract class, we firstly need to import the abc module. Let us look at an example.

```
from abc import ABC
class abs_class(ABC):
     #abstract methods
```

encapsulation in Python

Encapsulation is one of the four fundamental concepts in object-oriented programming including abstraction, encapsulation, inheritance, and polymorphism.

Encapsulation is the packing of data and functions that work on that data within a single object. By doing so, you can hide the internal state of the object from the outside. This is known as **information hiding**.

encapsulation example

```
class Counter:
    def __init__(self):
        self.current = 0

    def increment(self):
        self.current += 1

    def value(self):
        return self.current

    def reset(self):
        self.current = 0
```