OOPS

Abstraction I'm a coffee addict. So, when I wake up in the morning, I go into my kitchen, switch on the coffee machine and make coffee. Sounds familiar?

Making coffee with a coffee machine is a good example of abstraction.

You need to know how to use your coffee machine to make coffee. You need to provide water and coffee beans, switch it on and select the kind of coffee you want to get.

The thing you don't need to know is how the coffee machine is working internally to brew a fresh cup of delicious coffee. You don't need to know the ideal temperature of the water or the amount of ground coffee you need to use.

Someone else worried about that and created a coffee machine that now acts as an abstraction and hides all these details. You just interact with a simple interface that doesn't require any knowledge about the internal implementation.

only exposing high-level public methods for accessing an object

Tow types

1) **Data Abstraction**
2) **Process Abstraction**

Encapsulation

In general, encapsulation is a process of wrapping similar code in one place

we can bundle data members and functions that operate together inside a single class.

containing information in an object, exposing only selected information

Class

A class can be understood as a template or a blueprint, which contains some values, known as data members, and some set of rules, known as behaviors or methods. The data and methods that are defined in the class are automatically taken when an object is created. The class is a template or blueprint for objects. One can make as many objects as they want to be based on a class.

Object

Object is instance of class

Interface An interface defines the syntax that any entity must adhere to

Inheritance

child classes inherit data and behaviours from the parent class

Polymorphism

Many methods can do the same task