

## Part02

إمتى وليه تستخدم كل واحد فيهم؟ Abstract Class vs Interface:

من أكثر المفاهيم التي بتعمل لخطة لينا كـ Developers هي الفرق بين Abstract Class و Interface.

رغم إن الاثنين بيحققوا مبدأ التجريد (Abstraction)، بس الاستخدام بينهم مختلف تماماً لو فهمنا كل واحد فيهم.

عشان تختار صح، اسأل نفسك سؤال واحد بس: أنا بوصف "هو إيه" ولا بوصف "بيقدر يعمل إيه"؟

تعالوا نبسطها:

Abstract Class represents "IS-A" relation

الـ Abstract Class بنستخدمه لما نكون بنيني هيكل وراثه (Hierarchy) لحاجات مترابطة ببعضها.

مثال:

تحيل عندنا كلاس اسمه Animal.

الـ Dog is an Animal

الـ Cat is an Animal

هنا نستخدم Abstract Class، لأن الكلب والقطه "ورثوا" صفات مشتركة (زي التنفس، النوم) من فصيل الحيوانات، ويمكن نكتب كود مشترك بينهم في الـ Base Class ده.

Interface represents "CAN-DO" relation

الـ Interface عبارة عن (Contract) أو قدرة معينة، ملوش علاقة الكلاس ده نوعه إيه، المهم إنه بيقدّر ينفذ الوظيفة دي.

1-

مثال:

تحيل عندنا Interface اسمه IFlyable.

الـ Bird can Fly

الـ Airplane can Fly

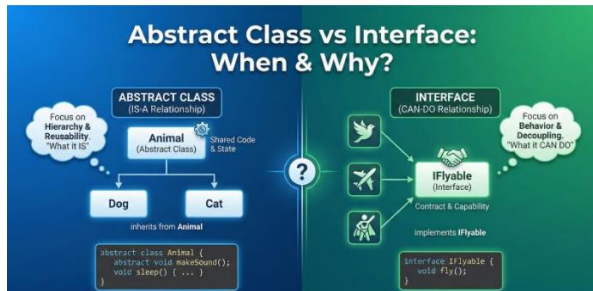
هنا كلهم مشتركين في "قدرة" معينة وهي الطيران. فنستخدم Interface لأننا بنوصف سلوك مش أصل الكائن.

استخدم Abstract Class: لما يكون عندك كلاسات ليها نفس "الأصل" وعاوز تشارك كود بينهم (Reusability).

استخدم Interface: لما تكون عاوز توصف "وظيفة" ممكن أي كلاس يعملها بغض النظر عن هو مين (Decoupling).

[SoftwareEngineering](#) [#OOP](#) [#CleanCode](#) [#ProgrammingTips](#) [#CodingLife](#) [#TechCommunity](#)

Show translation



## **2- What we mean by coding against interface rather than class ? and if u get it so What we mean by code against abstraction not concreteness ?**

Coding against an interface means relying on an abstraction rather than a concrete class, making the code more flexible and maintainable.

Coding against abstraction focuses on what an object does, not how it does it, reducing dependency on specific implementations.

## **3- What is abstraction as a guideline and how we can implement this through what we have studied ?**

Abstraction is a design guideline that focuses on exposing only essential behaviors of an object while hiding implementation details. It can be implemented using interfaces, abstract classes, and encapsulation through properties.

## **4-what is operator overloading?**

Operator overloading allows you to redefine how operators like +, -, , == work for your own classes, enabling objects to behave like built-in types for arithmetic or comparison.