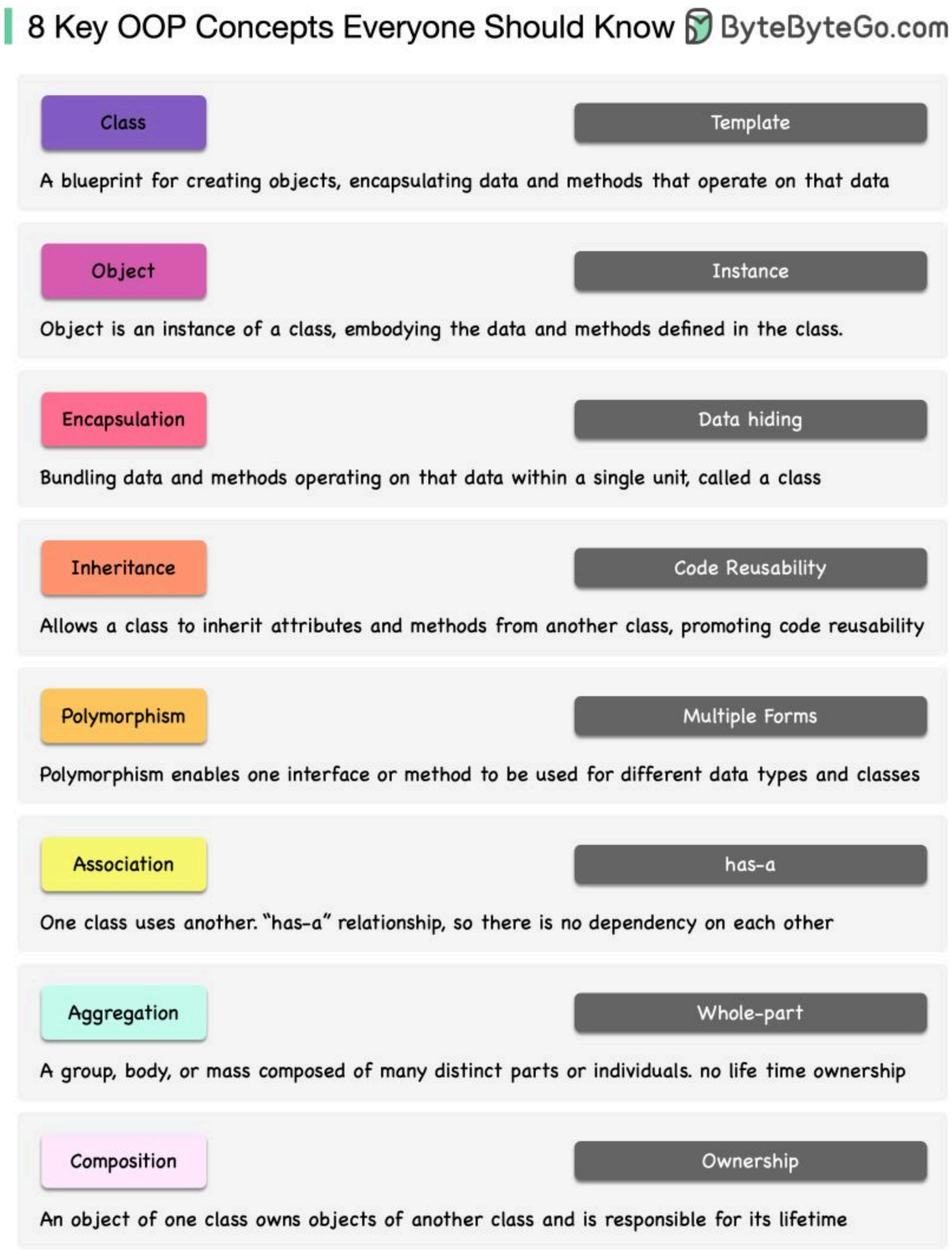
8 Key OOP Concepts Every Developer Should Know

Object-Oriented Programming (OOP) has been around since the 1960s, but it really took off in the 1990s with languages like Java and C++.



8 Key OOP Concepts Everyone Should Know

ByteByteGo.com

Class	Template	A blueprint for creating objects, encapsulating data and methods that operate on that data
Object	Instance	Object is an instance of a class, embodying the data and methods defined in the class.
Encapsulation	Data hiding	Bundling data and methods operating on that data within a single unit, called a class
Inheritance	Code Reusability	Allows a class to inherit attributes and methods from another class, promoting code reusability
Polymorphism	Multiple Forms	Polymorphism enables one interface or method to be used for different data types and classes
Association	has-a	One class uses another. "has-a" relationship, so there is no dependency on each other
Aggregation	Whole-part	A group, body, or mass composed of many distinct parts or individuals. no life time ownership
Composition	Ownership	An object of one class owns objects of another class and is responsible for its lifetime

Why is OOP Important? OOP allows you to create blueprints (called classes) for digital objects, and these objects know how to communicate with one another to make amazing things happen in your software. Having a well-organized toolbox rather than a jumbled drawer of tools makes your code tidier and easier to change.

In order to get to grips with OOP, think of it as creating digital Lego blocks that can be combined in countless ways. Take a book or watch some tutorials, and then practice writing code - there's no better way to learn than to practice!

Don't be afraid of OOP - it's a powerful tool in your coder's toolbox, and with some practice, you'll be able to develop everything from nifty apps to digital skyscrapers!