



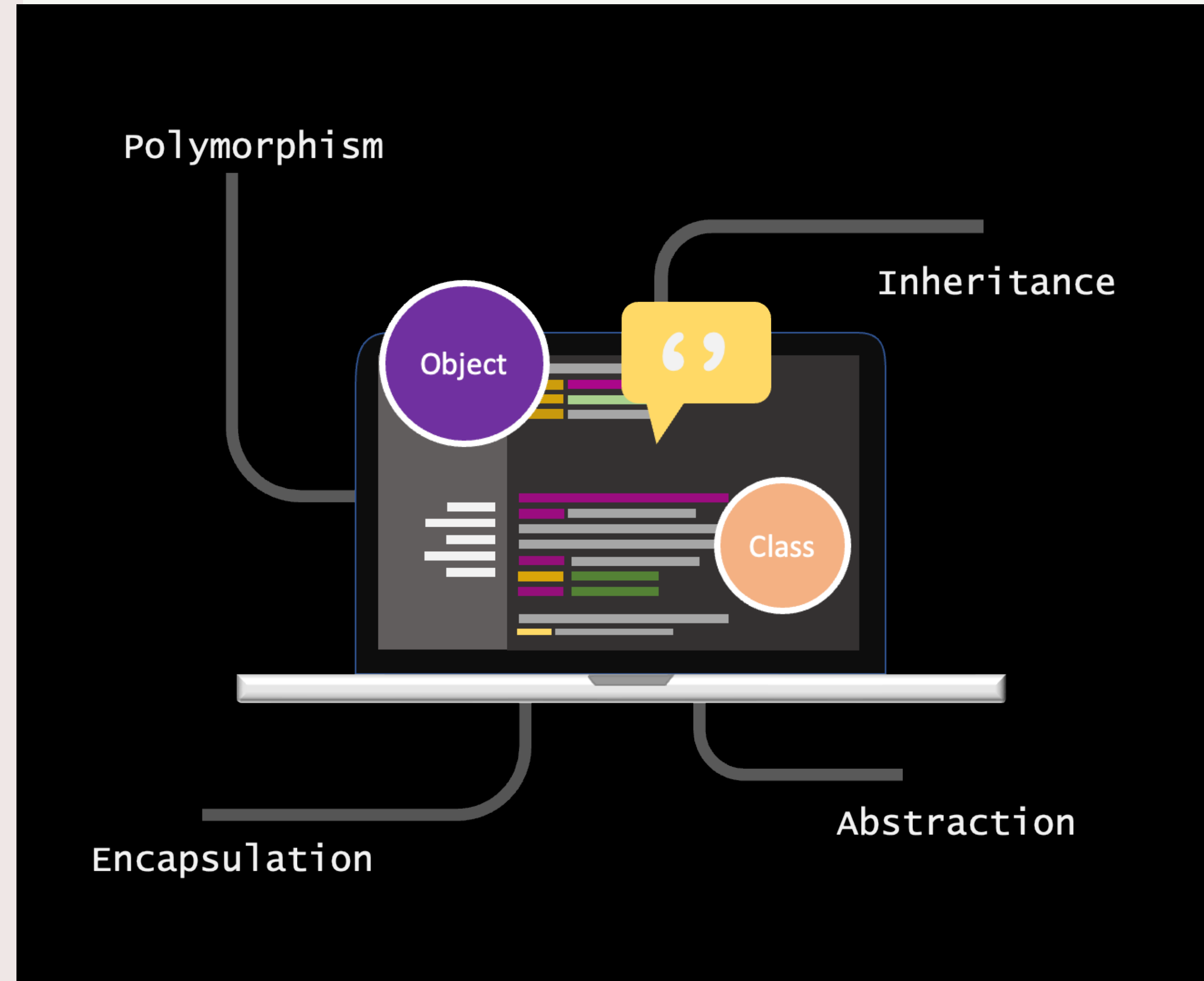
OBJECT ORIENTED PROGRAMMING

OOP

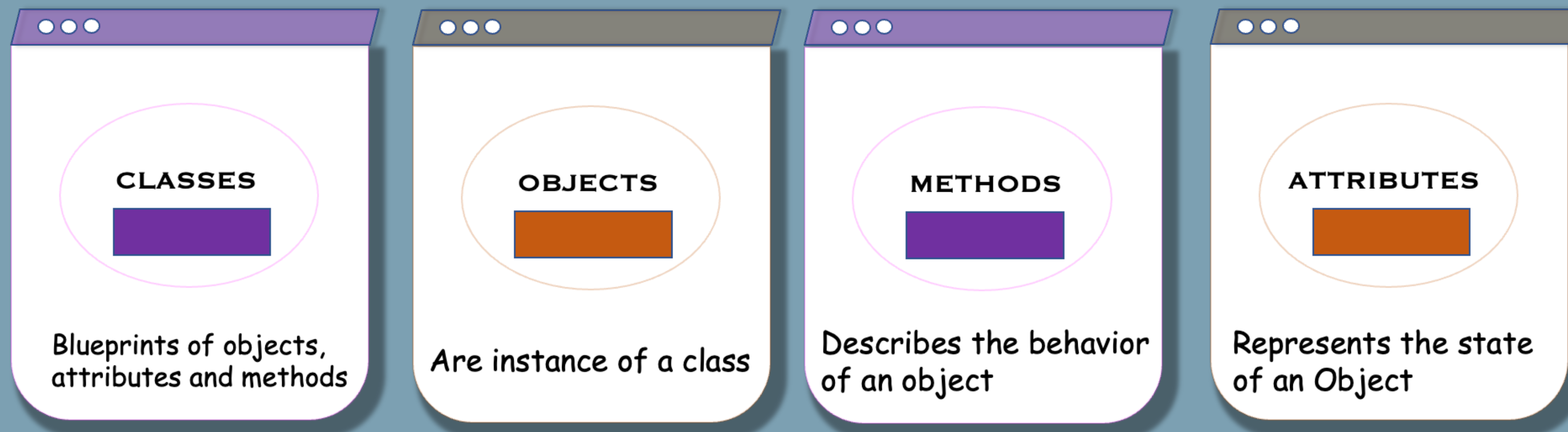
OBJECT ORIENTED PROGRAMMING

What is Object Oriented Programming (OOP)?

- It is a programming paradigm in computer science that relies on the concept of classes and objects
- It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects.



FOUR BUILDING BLOCKS OF OOP

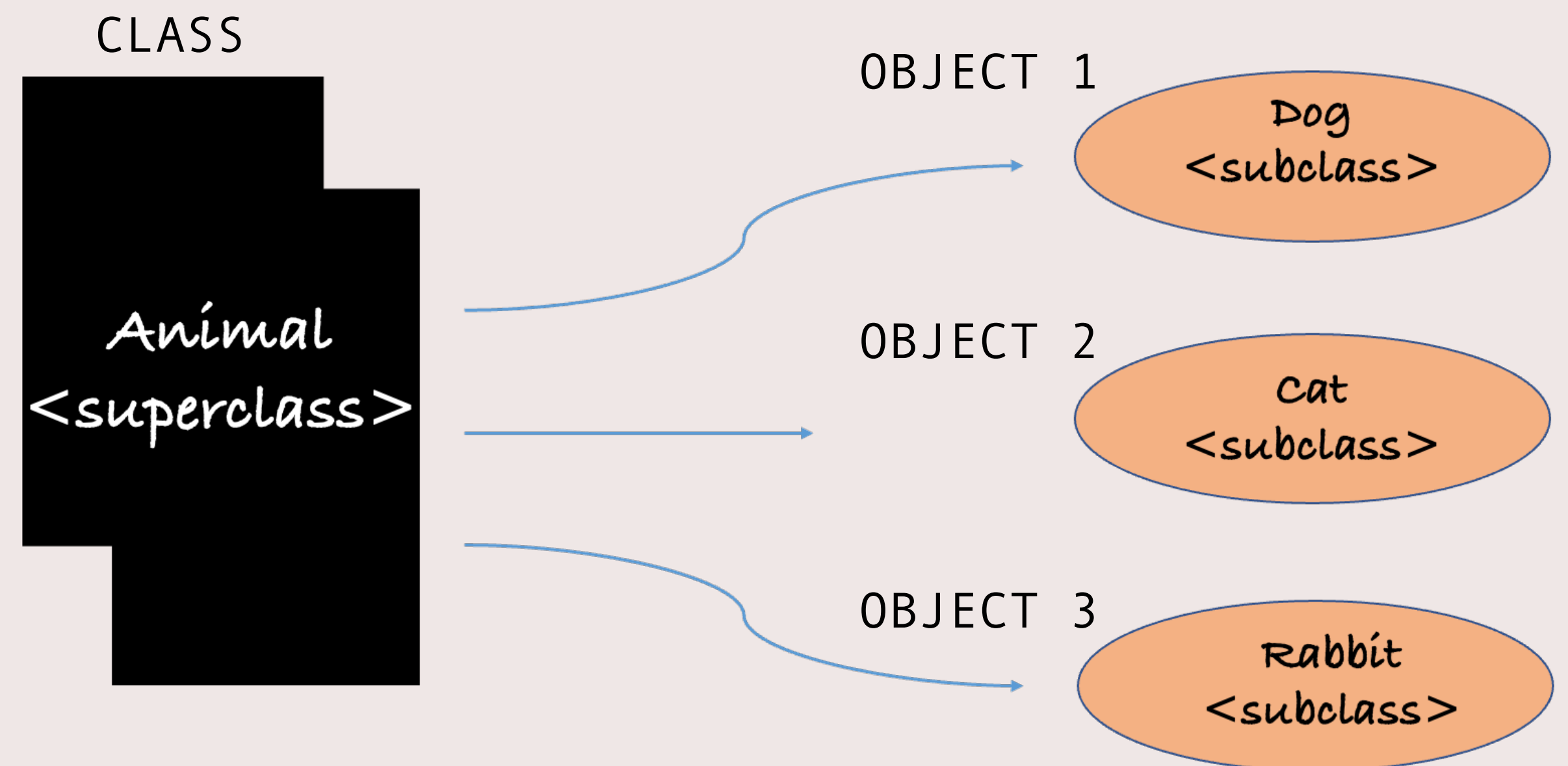


CLASSES

A class is an abstract blueprint for creating specific objects with shared attributes. It represents broad categories like Car or Dog. Classes define the attributes that instances of the class will have, but not their specific values.

OBJECTS

Objects are, unsurprisingly, a huge part of OOP! Objects are instances of a class created with specific data. For example, in the illustration by the right, Dog is an instance of the Animal class.

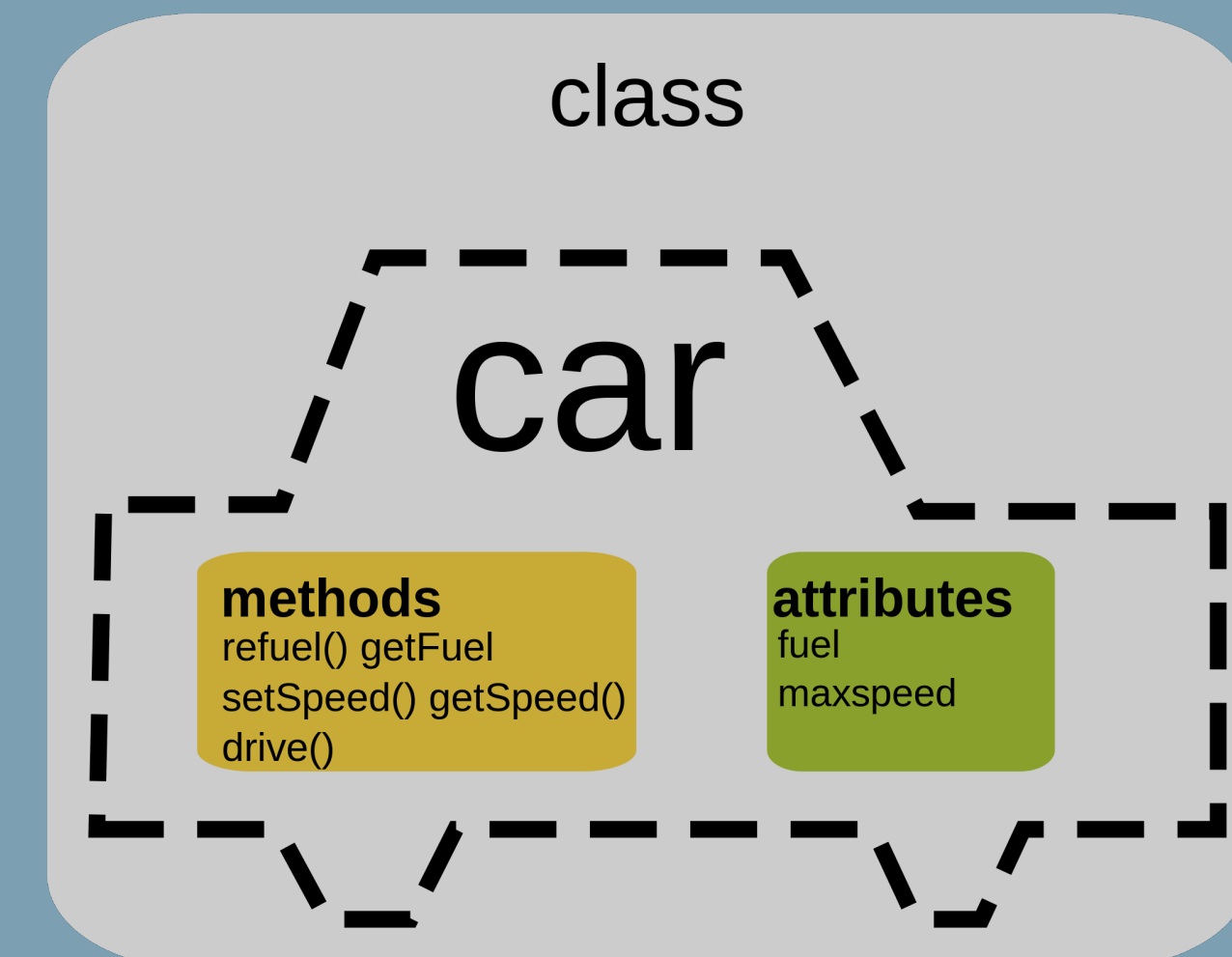


METHODS

Methods in a class perform actions and can provide information or update object data. They are defined in the class and help with reusability and encapsulation. This simplifies debugging as errors can be found and fixed in one place.

ATTRIBUTES

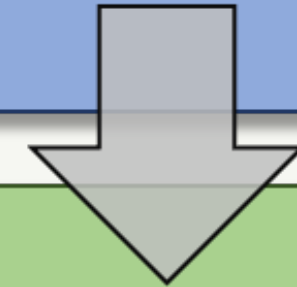
These are defined in the class template and represent the state of an object. Objects contain data stored in the attribute field. For example, a puppy and a dog might be treated differently at a pet camp. The birthday could define the state of an object and allow the software to handle dogs of different ages differently.



FOUR PILLARS OF OOP

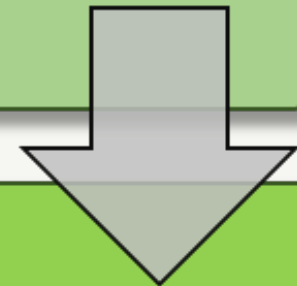
Abstraction

“Eliminate the Irrelevant,
Amplify the Essential”



Encapsulation

“HIDING THE UNNECESSARY”



Inheritance

“MODELLING THE SIMILARITY”



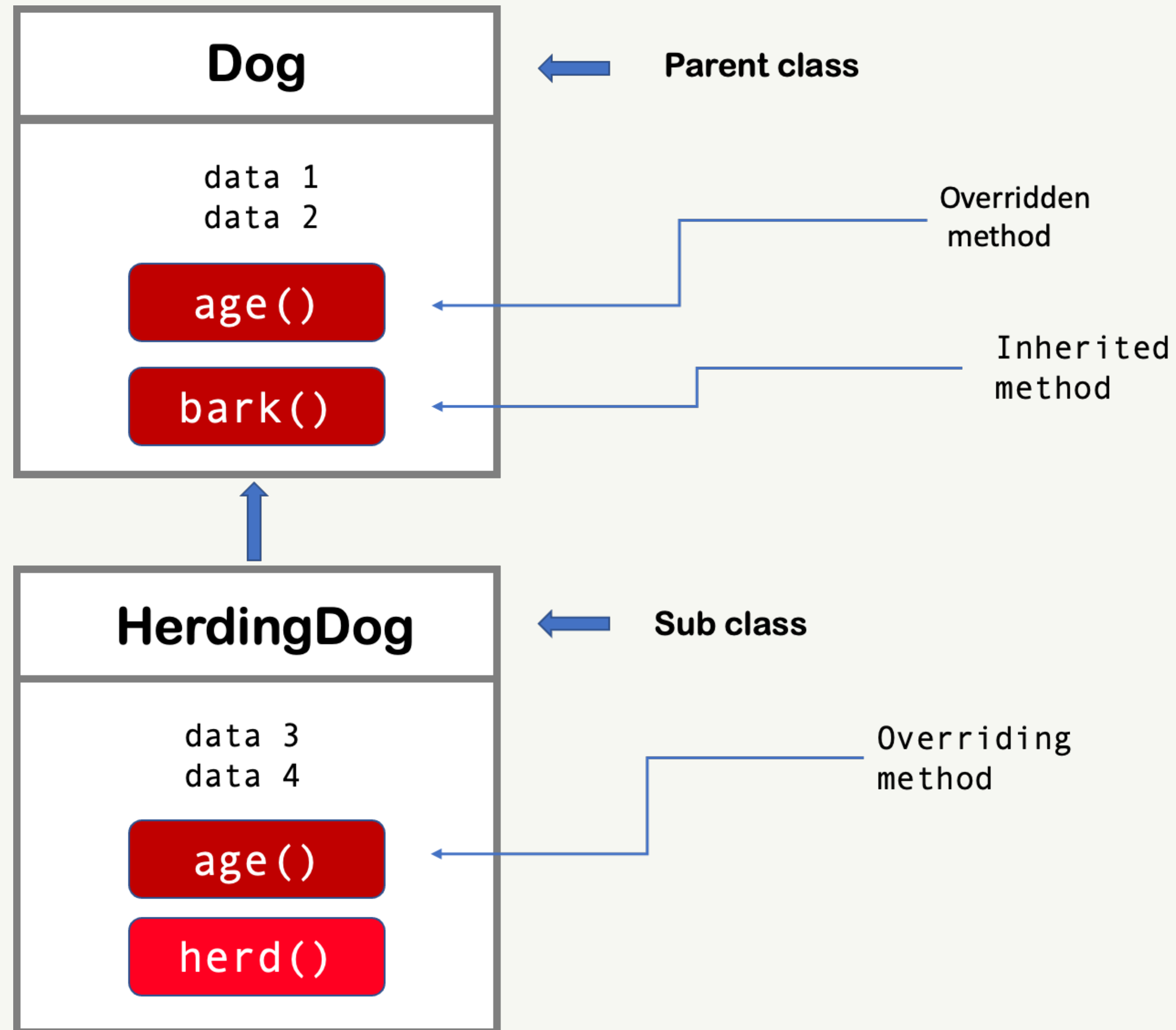
Polymorphism

“SAME FUNCTION DIFFERENT BEHAVIOR”

INHERITANCE

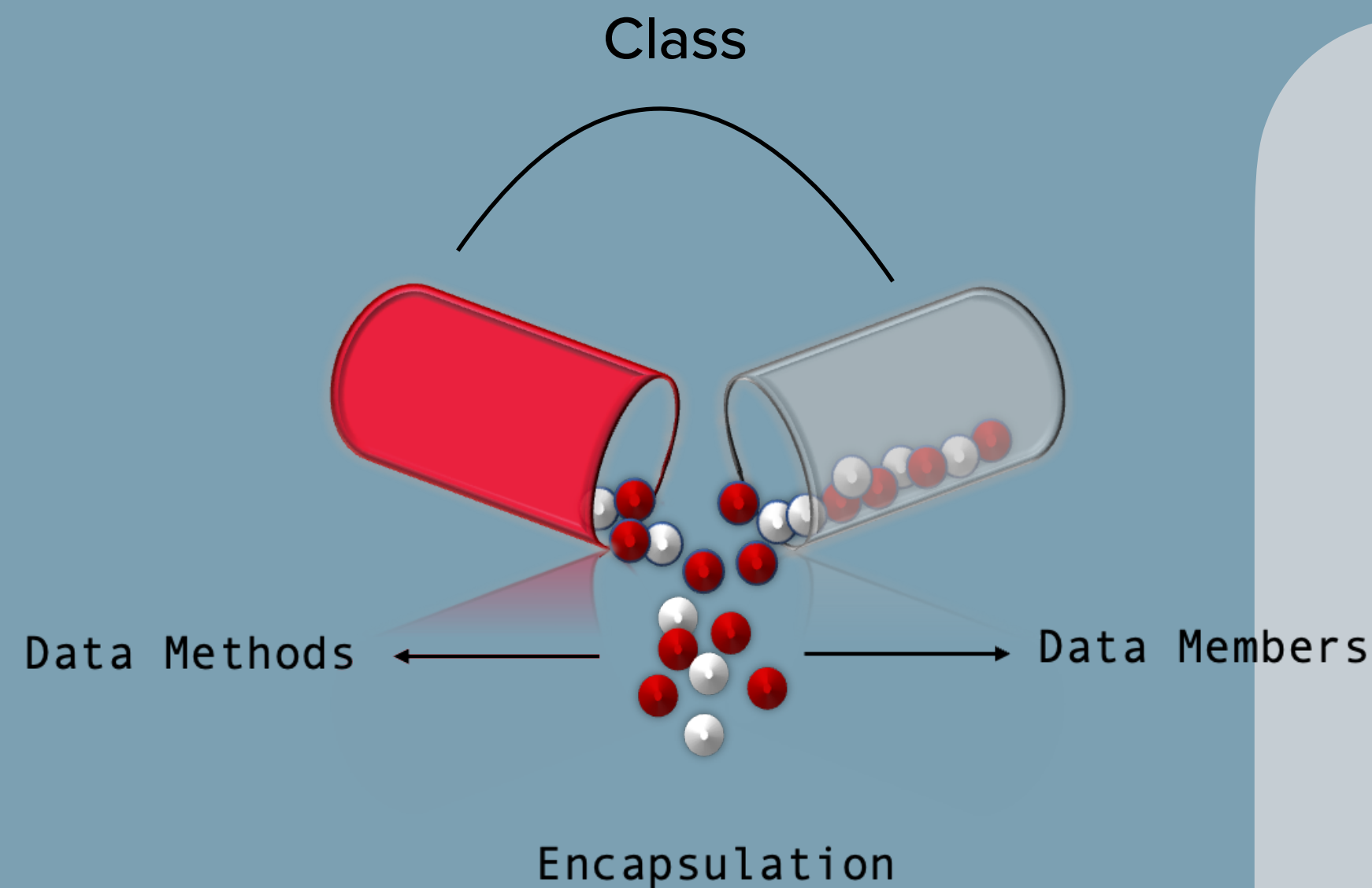
Inheritance allows classes to inherit attributes and behaviors from parent classes, promoting reusability. By extending the functionality of the parent class, child classes can add additional attributes and behaviors. For example, a herding dog, as a child class of a dog, can have the unique herd() behavior. Inheritance simplifies programming by creating a generic parent class and enabling child classes to automatically access the functionalities of their parent class without recreating the structure.

Note: Parent classes are also known as superclasses or base classes. The child class can also be called a subclass, derived class, or extended class.



Inheritance (object-oriented programming)

Encapsulation



Encapsulation means containing all important information **inside an object**, and only exposing selected information to the outside world. Attributes and behaviors are defined by code inside the class template.

- **Adds security:** Only public methods and attributes are accessible from the outside
- **Protects against common mistakes:** Only public fields & methods are accessible, so developers don't accidentally change something dangerous
- **Protects IP:** Code is hidden in a class; only public methods are accessible by the outside developers
- **Supportable:** Most code undergoes updates and improvements
- **Hides complexity:** No one can see what's behind the object's curtain!

Abstraction

- ✓ Full Name
- ✓ Address
- ✓ Contact Number
- ✓ Tax Information
- ✓ Favorite Food
- ✓ Favorite Movie
- ✓ Favorite Actor
- ✓ Favorite Band

okay. we
might not
need all
these
customer
information
for a banking
application

Abstraction involves using simplified classes to represent complex concepts. It allows users to focus on relevant and useful information, thereby managing complexity effectively.

A good example is explained below. The user being the bank personnel/HR.

In OOP applications, abstraction hides unnecessary details from the user, making projects more manageable. It simplifies development by breaking down the system into smaller components

Polymorphism

Polymorphism enables objects to share behaviors through inheritance, allowing specific behaviors to override shared parent behaviors. It provides two ways for a method to exhibit different behaviors: method overriding and method overloading.



arbabwaseer@gmail.com

Examples of OOP Languages

Technology and programming languages are evolving all the time. We have seen the rise of many langs under the OOP category, but **Simula** is credited as the first OOP language.

Programming languages that are considered pure OOP treat everything like objects, while the others are designed primarily with some procedural process.

Examples of OOP languages:

- | | |
|-----------|---------------------------|
| • Scala | • Python |
| • Emerald | • c++ |
| • Ruby | • JavaScript |
| • JADE | • Visual <u>basic.NET</u> |
| • Java | • PHP |

Benefits of OOP

- Code reusability through inheritance reduces redundancy.
- Improved integration with modern operating systems.
- Enhanced productivity through the use of libraries.
- Polymorphism enables adaptable functions.
- Easy upgrades and independent system functionality implementation.
- Encapsulation ensures self-contained objects, simplifying troubleshooting and collaboration.
- Encapsulation and abstraction hide complexity, facilitate software maintenance, and protect internet protocols.