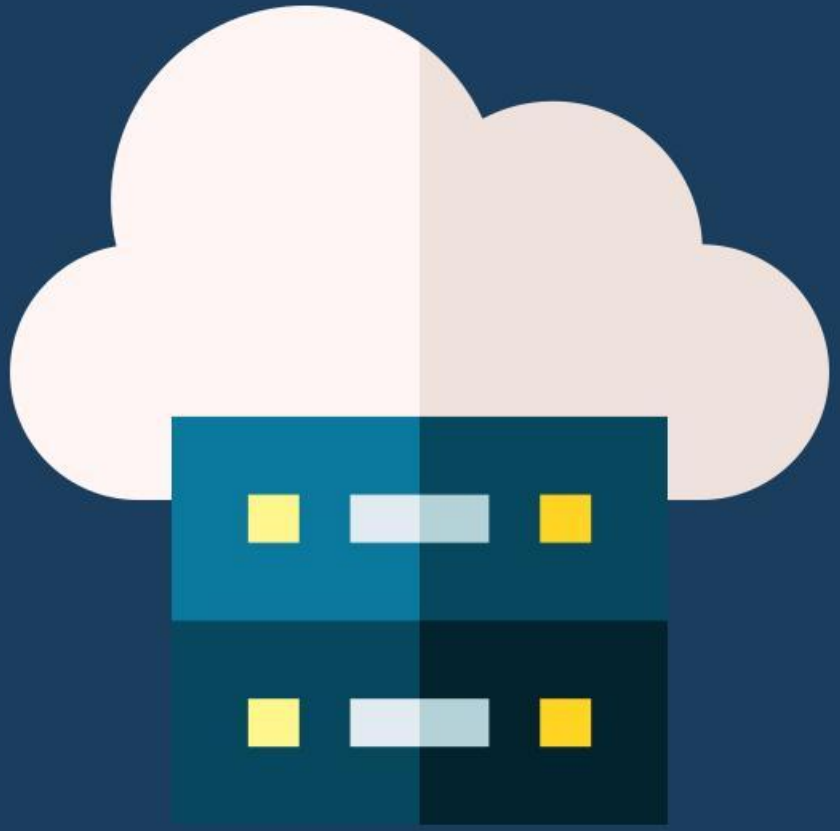


Object Oriented Programming

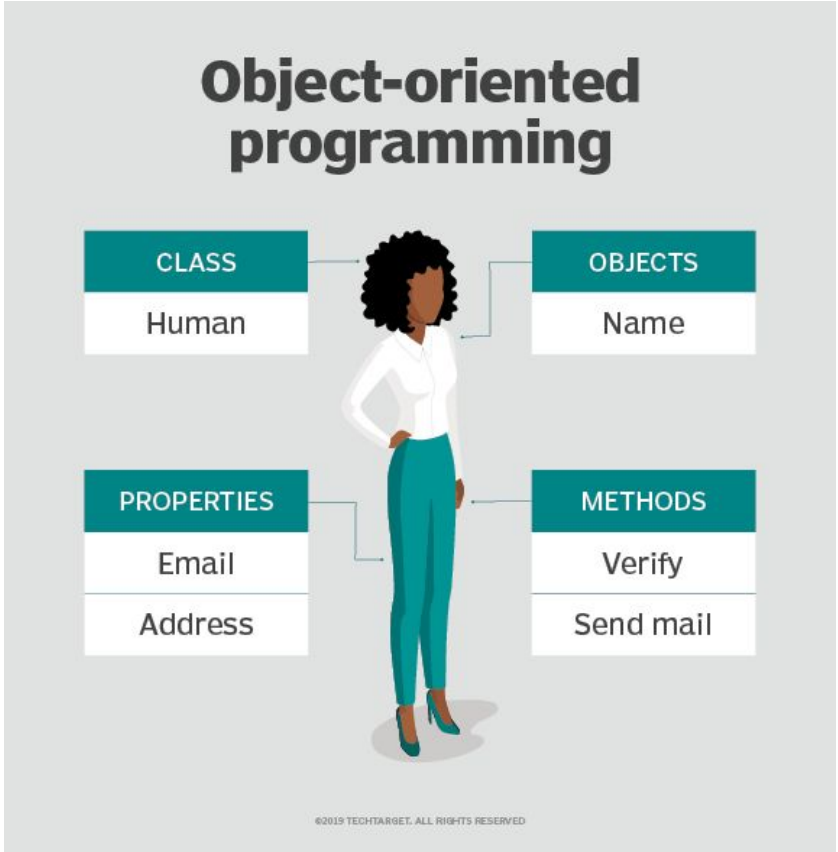
Presented by :
Murat, Olena, Yusuf, Ari

CONTENT

- HISTORY
- STRUCTURE
- OOP PRINCIPLES
- BENEFITS
- OOP LANGUAGES
- LIVE DEMO
- Q & A

WHAT IS OBJECT ORIENTED PROGRAMMING ?

- SIMPLE
- REUSABLE CODE BLOCKS



HISTORY

1960s

- **LISP** - Alan Kay
- **Sketchpad** - Ivan Sutherland
- **Simula**

1970s

1st version of the **Smalltalk** - Xerox PARC (ALan Kay, Dan Ingalls, Adele Goldberg)

1980s

Smalltalk and OOP to a wider audience being introduced - Adele Goldberg

1990s

Object-oriented programming developed as the dominant programming paradigm

Present

Python, Ruby, C#, VB.NET have emerged that are primarily object-oriented, but that are also compatible with procedural methodology

STRUCTURE of OOP —> How would you run a **petshop**?

You have **to keep many data**:

And there **many brands** and **diverse types** of each product …

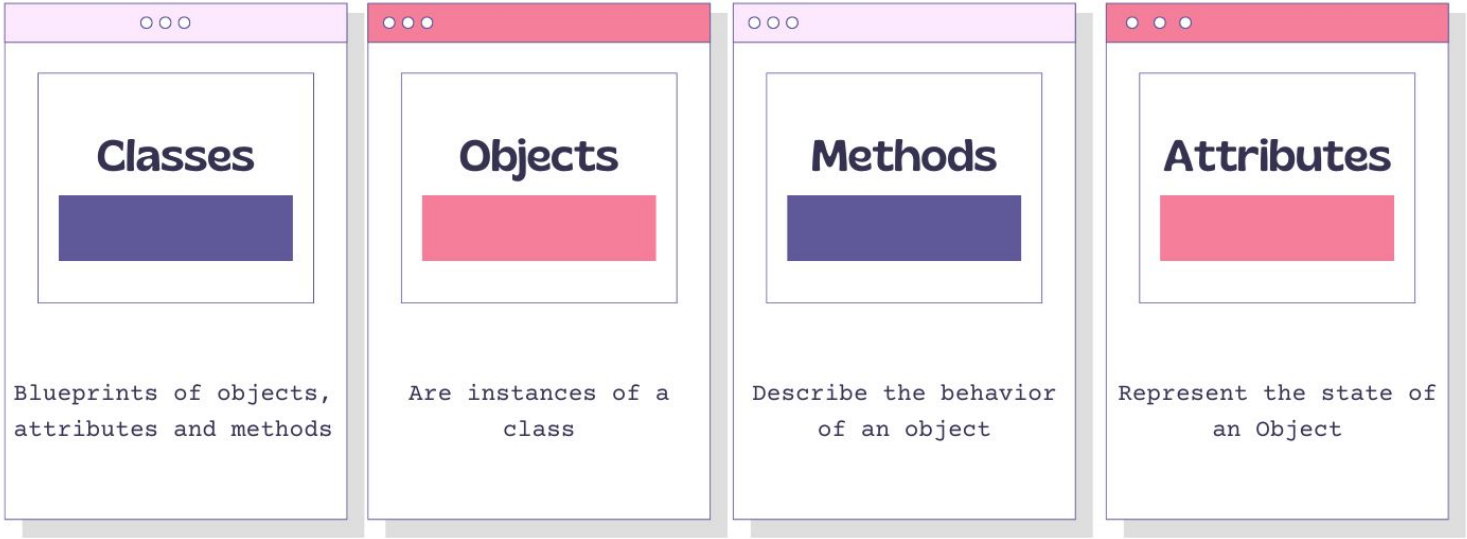
And to be able to **operate this data!**

We can write code for *each item*, and operate them *separately* - **huge** and **long code**!

OR

We can **group related information** together to produce **shorter** and more **reusable code!**

Structure of Object-Oriented Programming



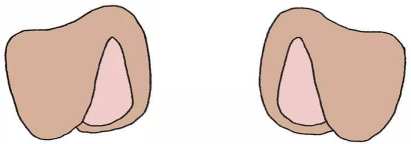
1. At first we need to create a **class**

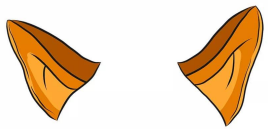
A ***class*** is like a *set of instructions*.

It describes the *methods (functions)*
and *properties (variables)*

that **will exist** in an object
when it's created.

2. Class template contains **attributes** which represent the state of an object

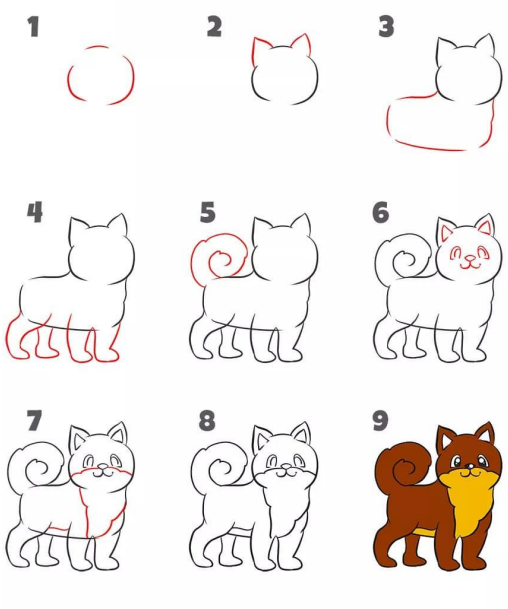


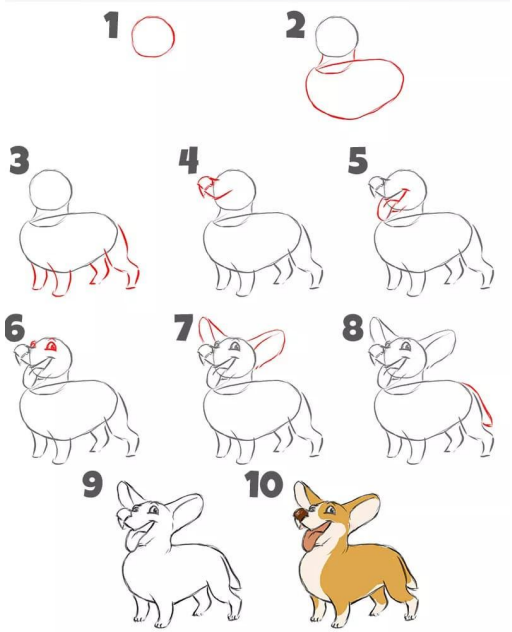


3. **Many objects** with *unique values* can be created from a **single definition**

Same *properties* (already exist in the class constructor)

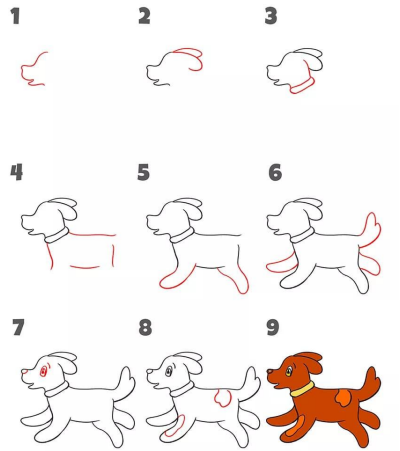
gain *unique* values result in unique objects.

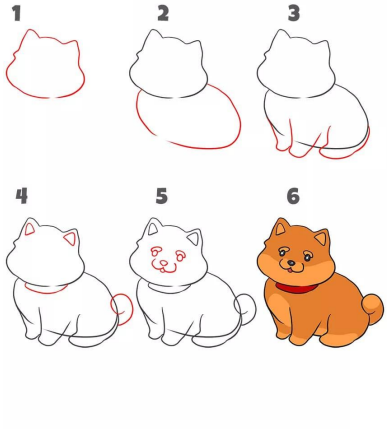




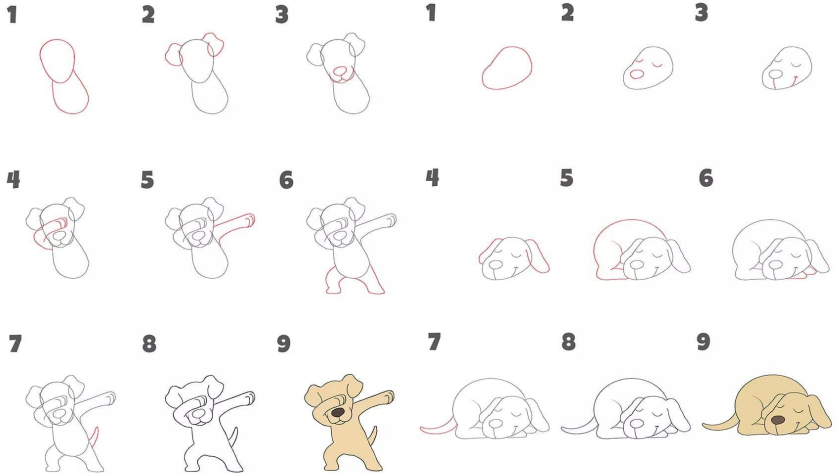
4. **Methods** are functions that are defined inside a class that describe the **behavior** of an object.

Different objects with different behavior





The same objects with different behavior



OOP PRINCIPLES

wrapping up of data
under a single unit

reducing cognitive load

acquiring the properties
from one class to
other classes

thinking about reuse

showing only
essential details
to users

defining conceptual
boundaries

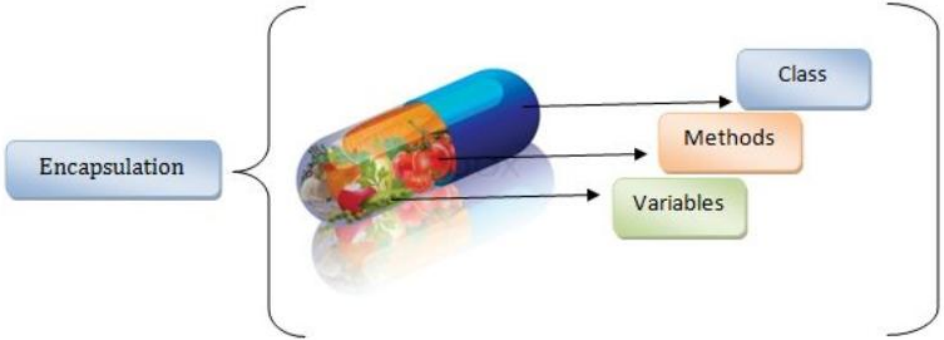
ability of an object to
be in many forms

managing internal
dependencies

ENCAPSULATION

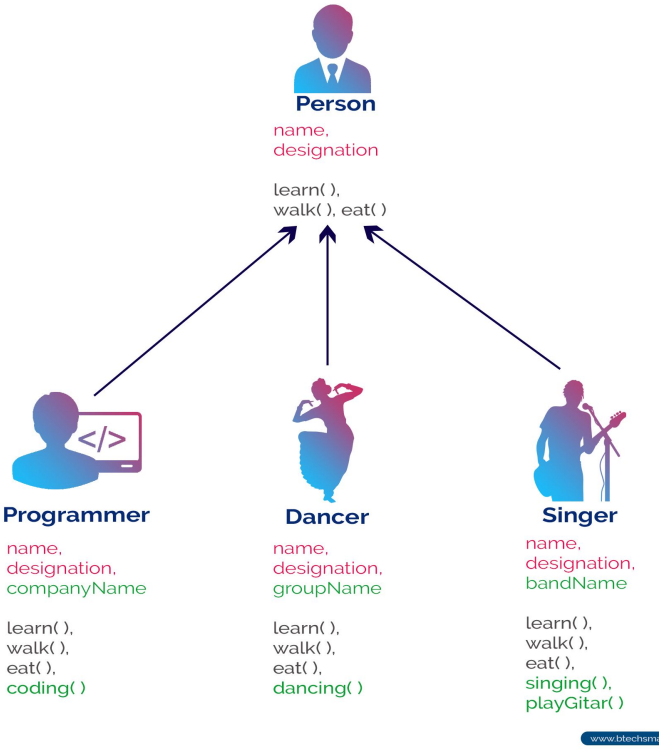
- Each object is privately held inside a class.
- Other objects do not have access to this class.
- Only able to call a list of public functions or methods.
- This provides greater program security and avoids unintended data corruption.

- Hide state, Reveal Behavior



INHERITANCE

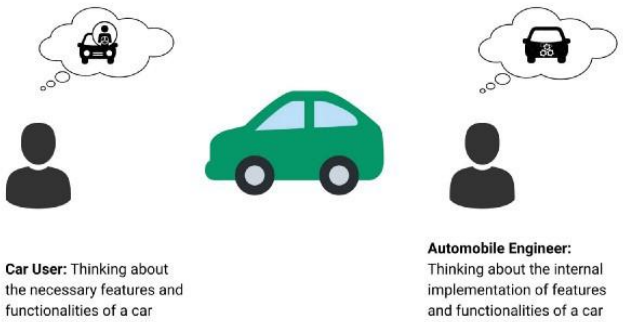
- Inheritance allows classes to inherit features of other classes.
- This property forces a more thorough data analysis.
- Inheritance reduces development time
- That ensures a higher level of accuracy.

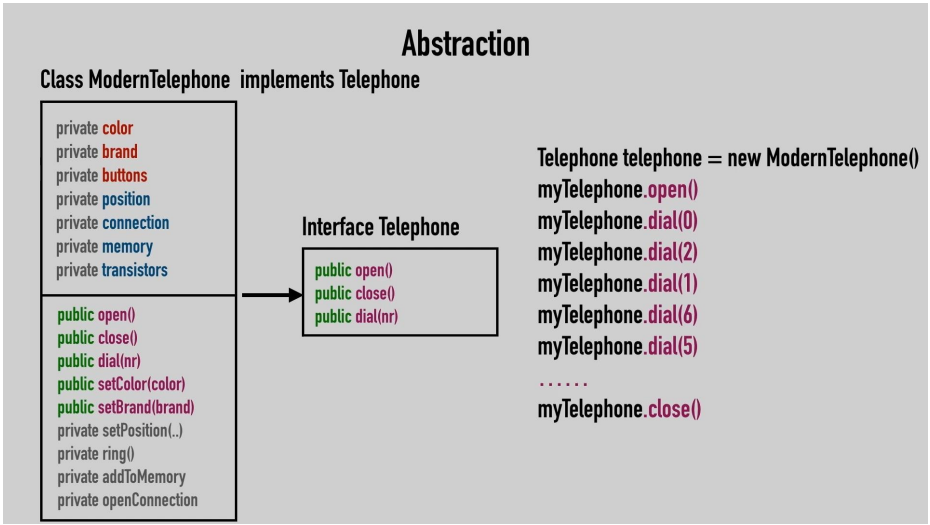


INHERITANCE

ABSTRACTION

- Abstraction is used to handle complexity by allowing to see only relevant and useful information (simplify).
- Hiding the inner details from the outside world.
- Know what it does, not how it does it.

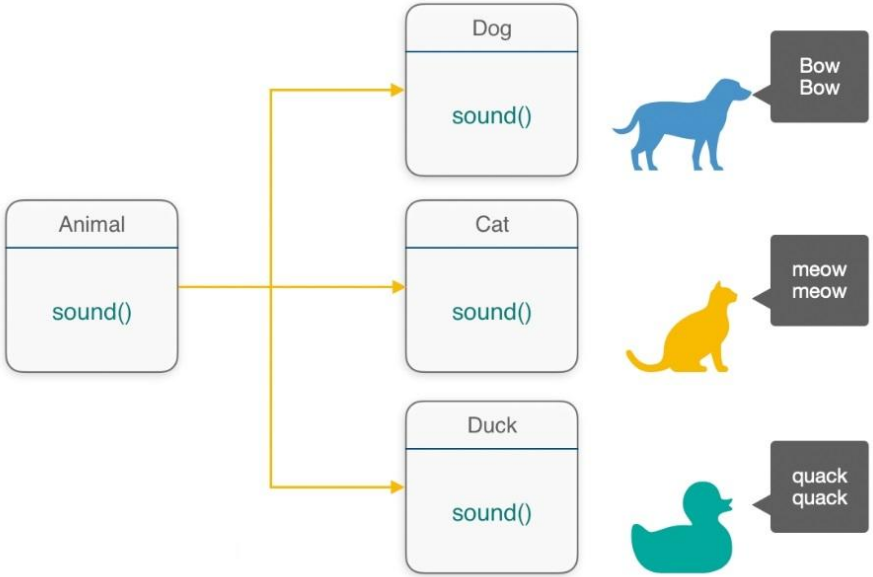




ABSTRACTION

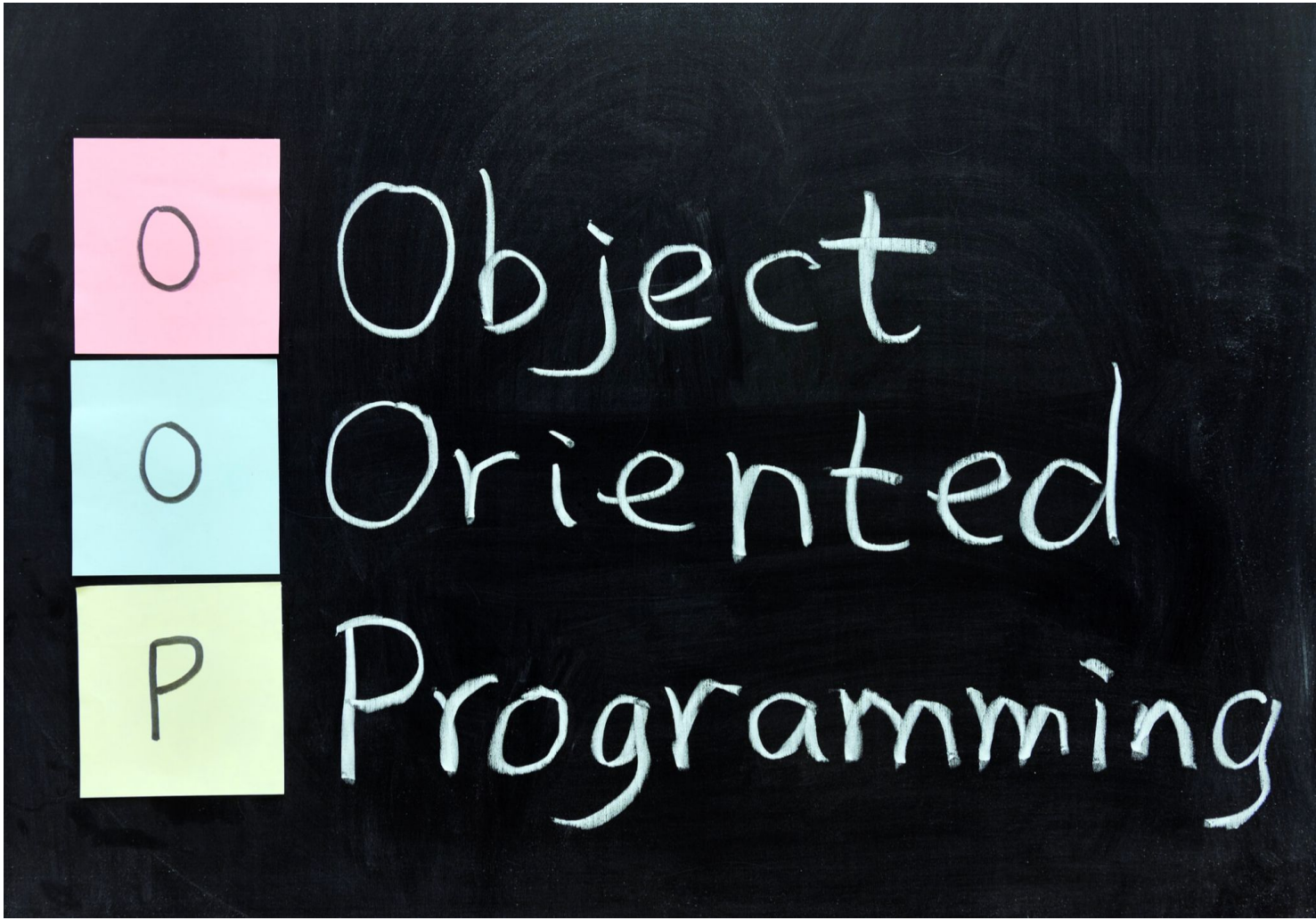
POLYMORPHISM

- Methods produce different outputs or do different jobs depending on the object.
- It is to give methods flexibility to behave polymorphously (instead of fixed tasks).
- The verb "to play" is used for all musical instruments but a piano is played differently from a cello



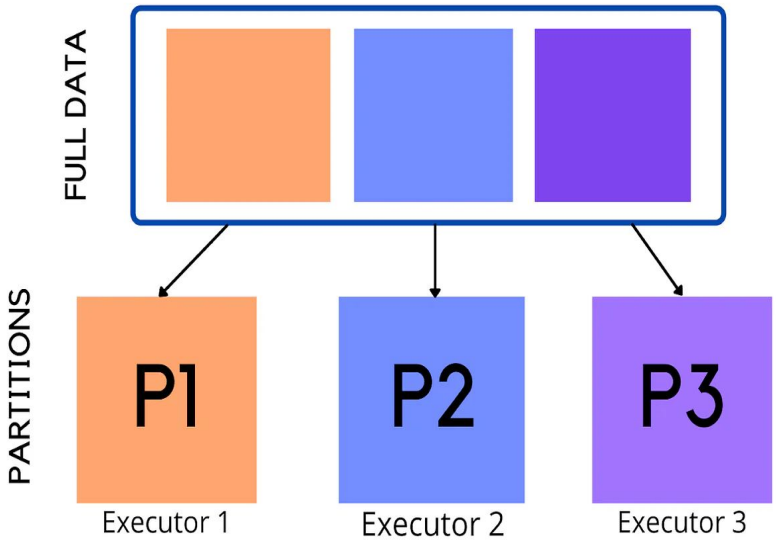
POLYMORPHISM

BENEFITS of the



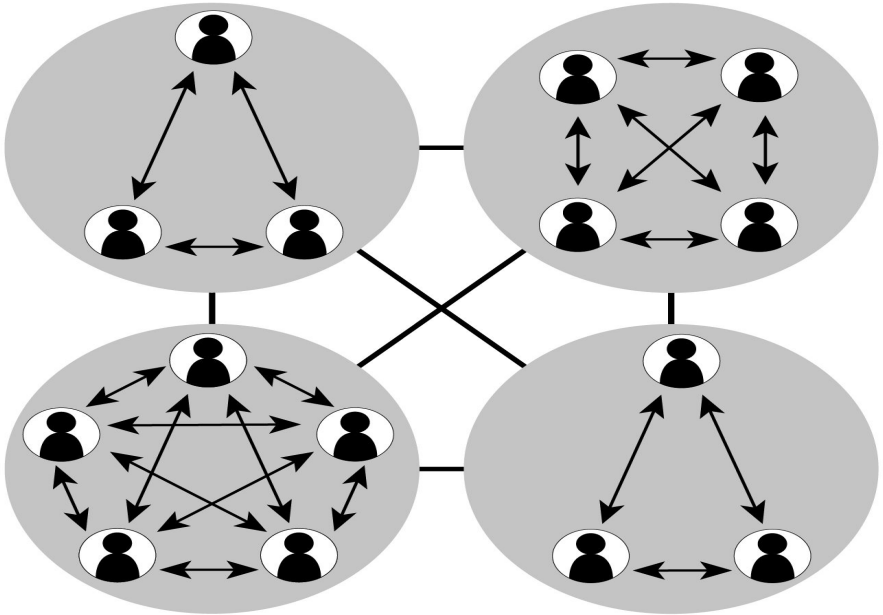
SIMPLICITY

- Easy to partition the work in a project based on objects



MODULARITY

- Building programs from standard working modules that communicate with one another



MODIFIABILITY

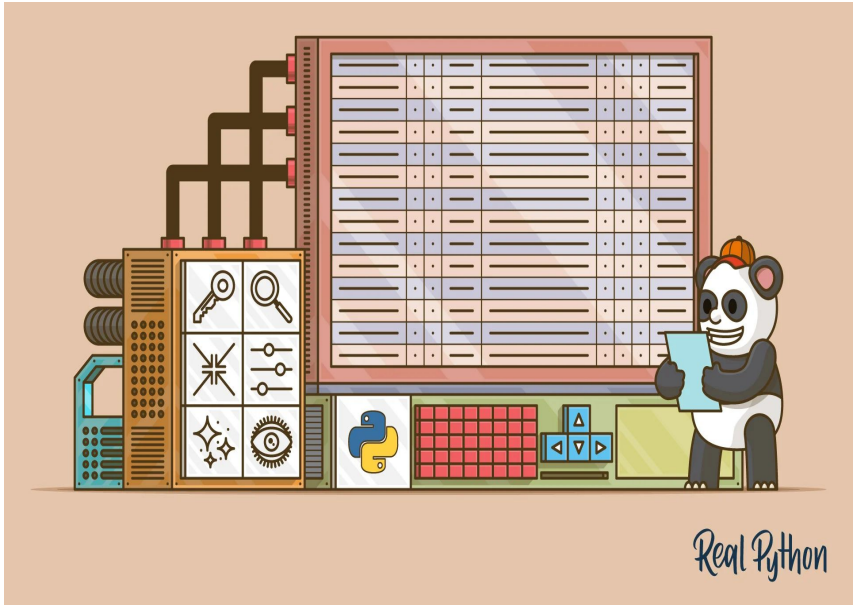
- Easy to make minor changes

RE-USABILITY

- Objects can be reused in different programs

INHERITANCE

- Eliminates redundant code and extends the use of existing classes

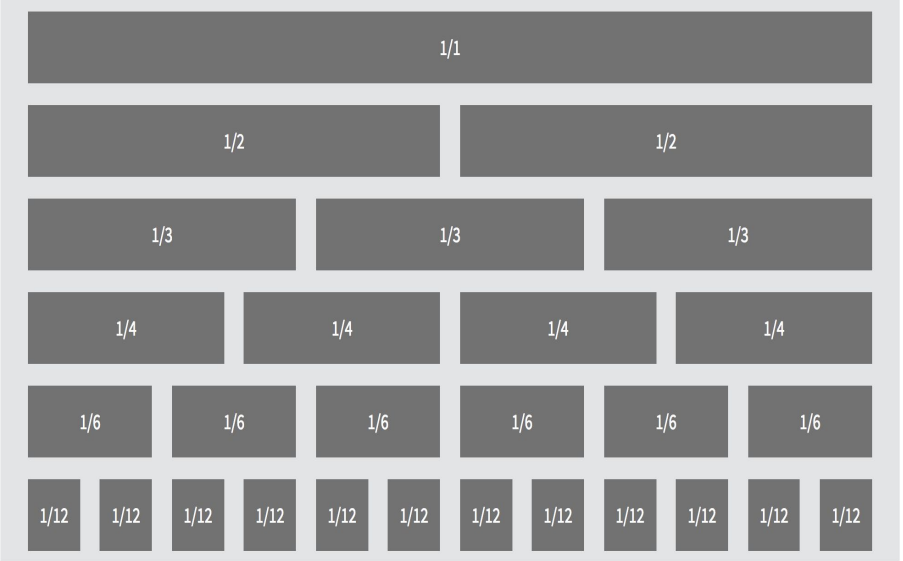


ABSTRACTION

- Helps programmer to build secure program

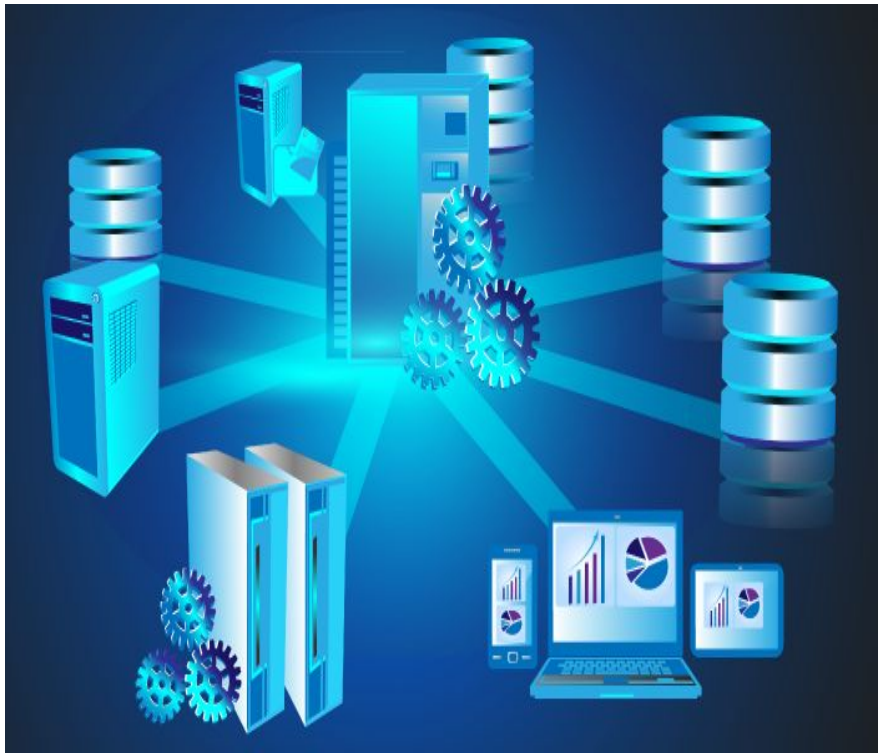
UPGRADATION

- Can be upgraded from small to large system



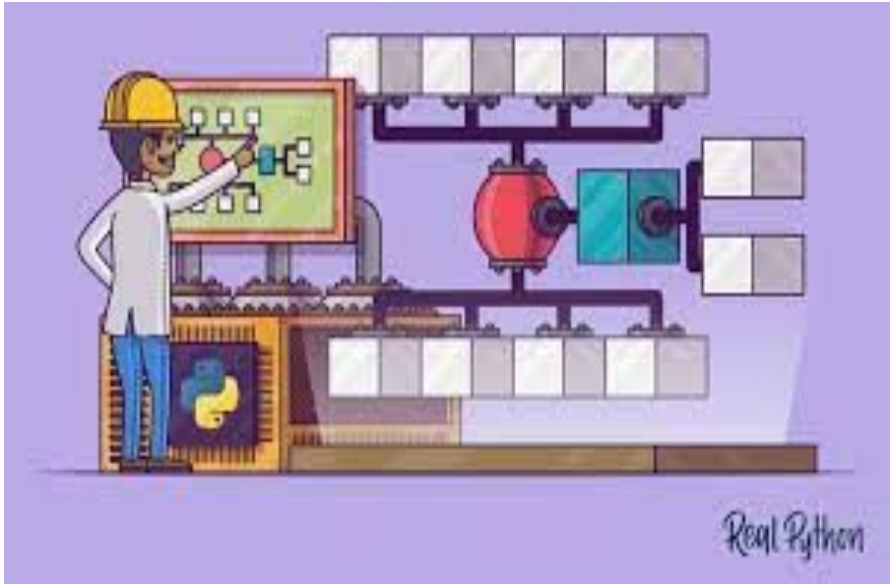
LOWER COST of DEVELOPMENT

- Reuse of software lowers the cost of the development



MAINTAINABILITY

- Objects can be maintained separately

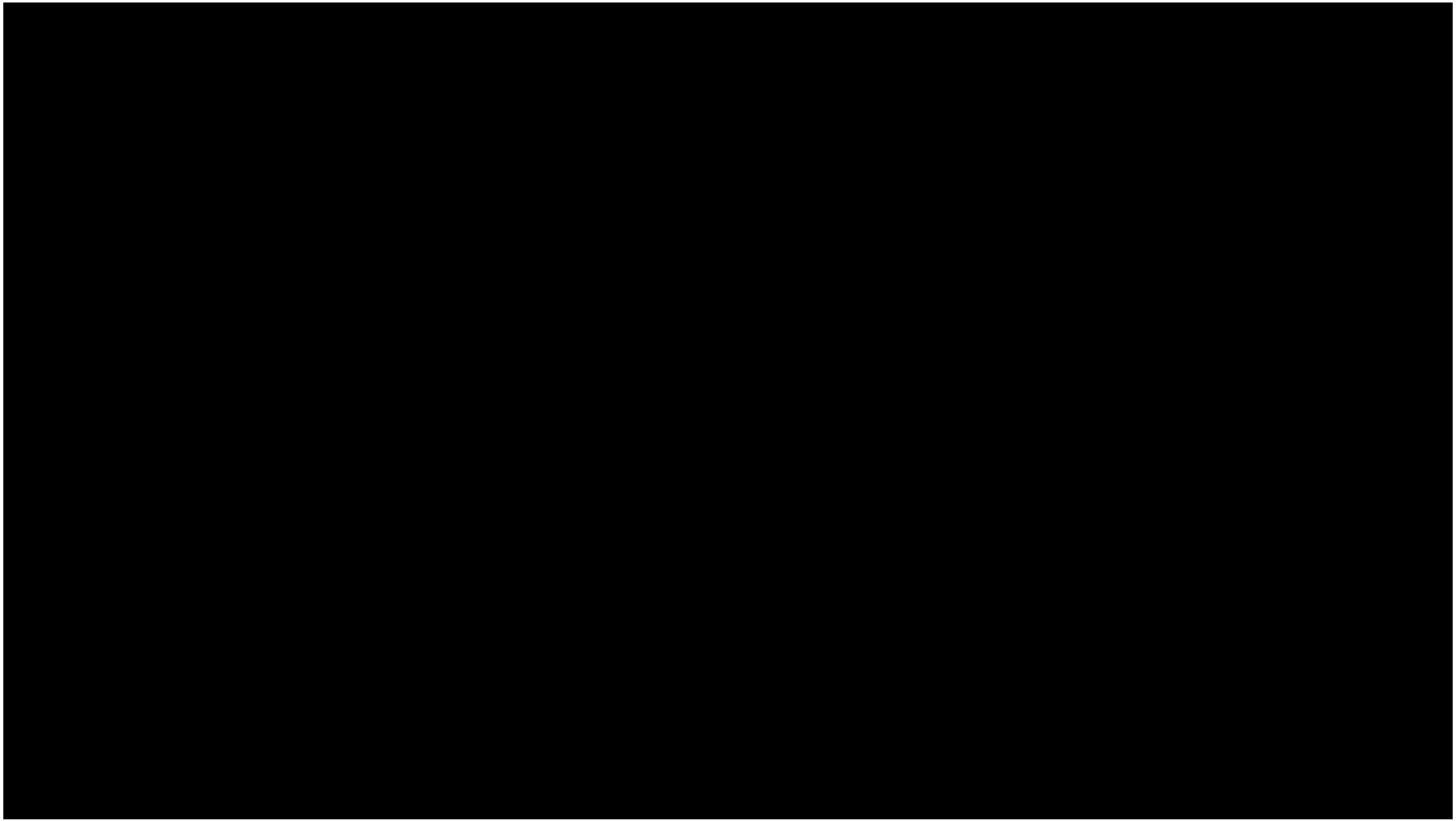


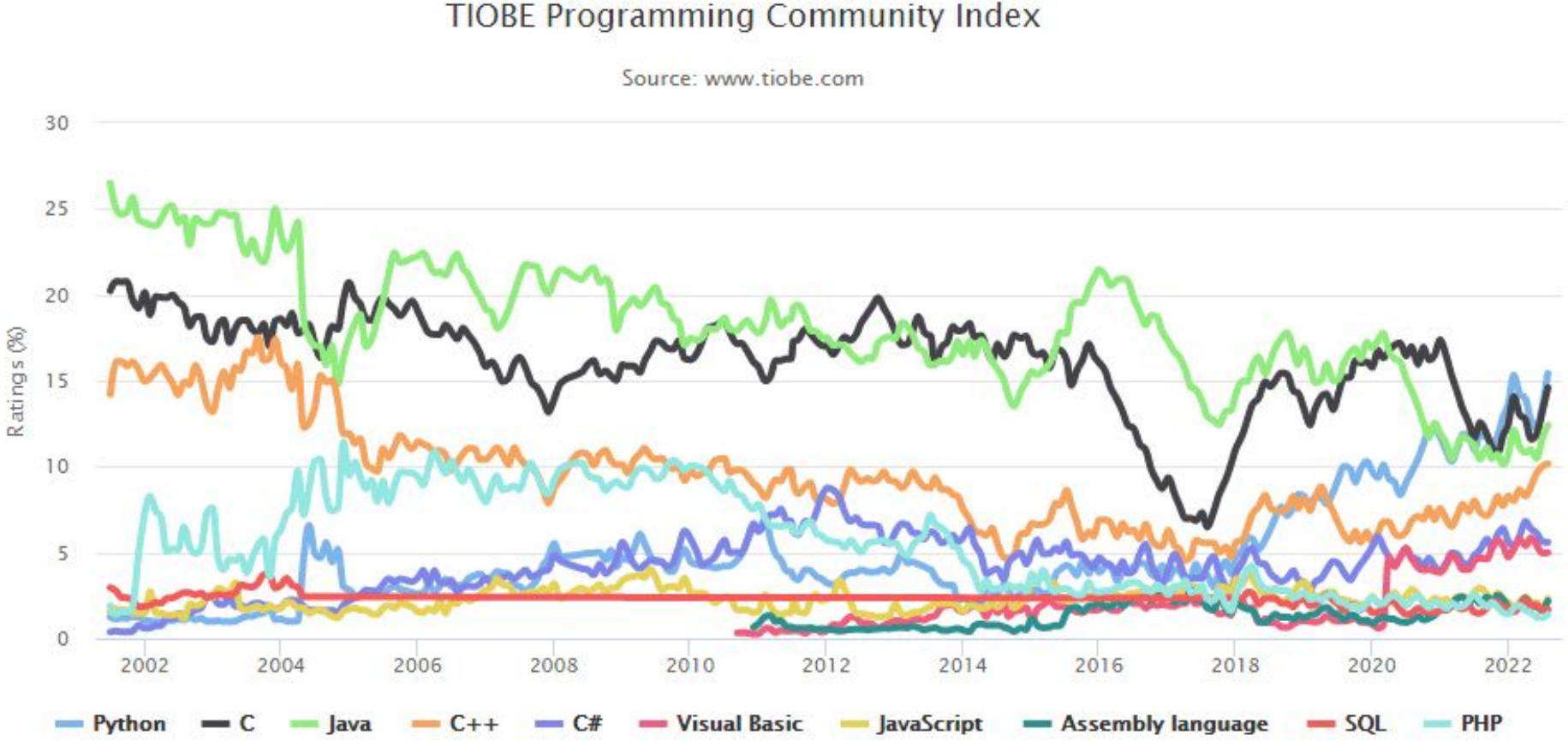
OOP LANGUAGES

- High Level
- More human readable
- Require translation by compiler
- Based on four principles
  - Encapsulation
  - Abstraction
  - Inheritance
  - Polymorphism.


Java






TIOBE Programming Community Index
Source: www.tiobe.com
Ratings (%)
30
25
20
15
10
5
0
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
Python
C
Java
C++
C#
Visual Basic
JavaScript
Assembly language
SQL
PHP


LIVE DEMO
<!--
text/javascript">
var code = "hack";
if(hack == "this") {
document.write("Hacked!!!</b
} else if(hack == "user") {
document.write("Access Grante
} else if(hack == "admin") {
document.write("Password
} else {
document.write("Unknown Reso
}
//-->
</script>
COMPUTER HACKED</p>


Q&A


THANK YOU