

# Lecture 01: Introduction to Object Oriented Programming

Swakkhar Shatabda

B.Sc. in Data Science  
Department of Computer Science and Engineering  
United International University

January 25, 2024



# Your Course Teacher

- Professor Dr. Swakkhar Shatabda
- Counselling Time
  - Sun/Wed 8:30 AM - 11:10 AM
- Expertise
  - Machine Learning
  - Optimization
  - Computational Biology
- Courses that I take:
  - Artificial Intelligence
  - Machine Learning
  - Reinforcement Learning
  - Deep Learning



# What is Data Science?

## Definition

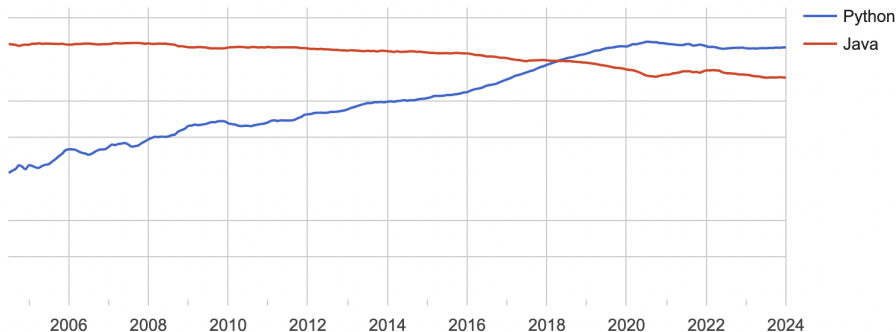
Data science is an **interdisciplinary** academic field that uses **statistics**, **scientific computing**, **scientific methods**, **processes**, **algorithms** and **systems** to **extract or extrapolate knowledge and insights** from **noisy**, **structured**, and **unstructured** data.

- A new paradigm - 'data driven' science.
- A **data scientist** is a professional who creates **programming** code and combines it with **statistical** knowledge to create **insights from data**.



# Why python?

PYPL Popularity of Programming Language



<https://pypl.github.io/PYPL.html>

<https://www.tiobe.com/tiobe-index/>



# Why Python?

```
[>>> import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *\*right\** now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!

# Why Python?

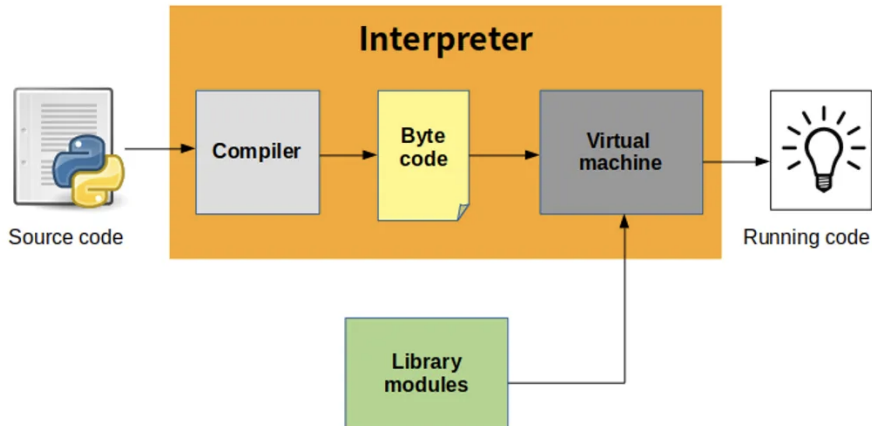
- It is platform-independent.
- It allows for fast development.
- It offers extensive libraries.
- It offers a more flexible approach to programming.
- It's free and open-source.

## Python!

Python is a **high-level**, **general-purpose** programming language. Its design philosophy emphasizes code **readability** with the use of significant **indentation**. Python is **dynamically typed** and **garbage-collected**. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming.



# How Python works?



# Two Paradigms

## Procedural Programming

- Process oriented, code that acts on data.
- What is happening?
- A program is a series of linear steps - C!
- As program grows larger - hard to manage!

## Object Oriented Programming

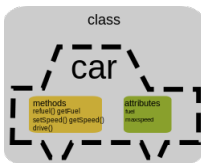
- data controls access to codes
- Who is being affected?
- Program is organized around its data - easy to manage / Java



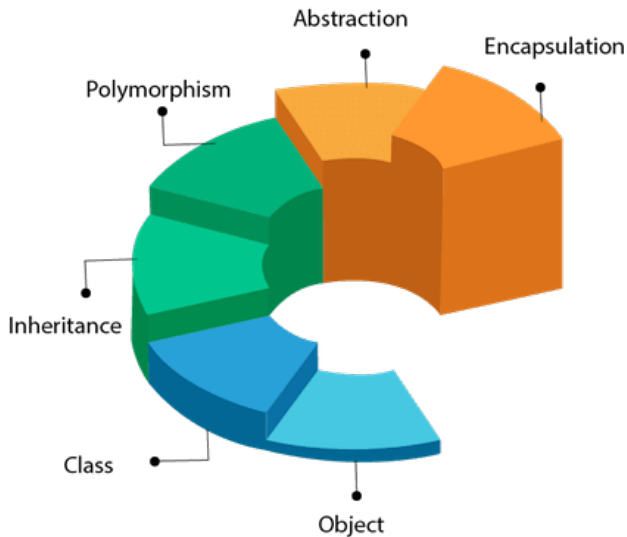


# Abstraction

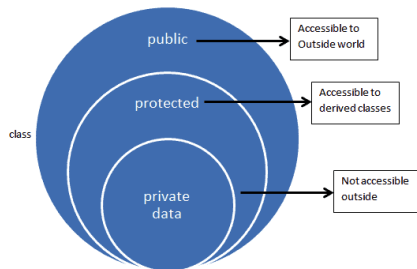
- Abstraction facilitates the easy conceptualization of real world objects into the software program.
- Humans manage complexity through abstraction.
- people do not think of a car as a set of tens of thousands of individual parts. They think of it as a well-defined object with its own unique behavior.
- A powerful way to manage abstraction is through the use of hierarchical classifications.



# OOP Principles



# Encapsulation



- Binds together code and the data it manipulates
- Keeps code and data safe from outside interference and misuse.
- Real Life Example: Car
- In Python: A class contains member variables and methods.
- Encapsulation guarantees the integrity of the data contained in the object.



# Inheritance

Motor Car

-speedLimit  
-startEngine()



Honda CRV

-infotainmentController  
-hFunction()



BMW 1-series

-xDriveDivider  
-bmwOperation()



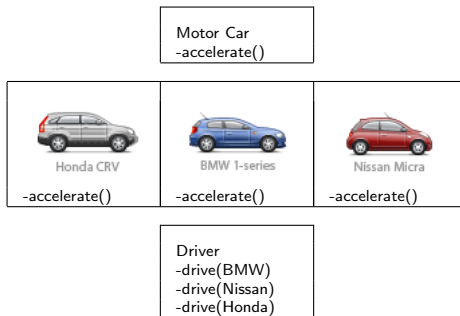
Nissan Micra

-extraSafetyBag  
-nFunction()

- Inheritance is the process by which one object acquires the properties of another object.
- An object needs to define those qualities that make it unique within its class and inherits general attributes from its parent.



# Polymorphism



- Polymorphism (from Greek, meaning “many forms”)
- One interface, multiple methods
- Think about a method that finds the length of strings and integers.



# Class and Object



- Class is a blueprint that defines some properties and behaviors.
- An object is an instance of a class that has those properties and behaviours attached.
- A class is not allocated memory when it is defined.
- An object is allocated memory when it is created.
- Class is a logical entity whereas objects are physical entities.



# Class in Python!

```
class Account:
    def __init__(self):
        self.name = ""
        self.age=0
    def show(self):
        print("Name:",self.name,"Age:",self.age)

ac = Account()
ac.name  = "Swakkhar"
ac.age = 41
ac.show()
```

