What is oop?

Object Oriented Programming

Combo of entities allows u to combo data elements and operational elements into single elements = class

Advantages:

Data and functions combined

Better modularity

Low coupling

Re-use easier

Closely modelling the rl world

4 principles of oop:

Abstraction – description of classes

Encapsulation – how does this work? Packaging or enclosing things of interest into one entity

Inheritance – when a class derives from another class

Polymorphism – individual dif classes can be given same \_ and be used dif depending on method. Tell method to do smth and it will know how to do it.

What is instantiation?  
What is initialisation?

Abstraction:

e.g. car. Abstract understanding of a car e.g. doors, windows etc. shared understanding of what a car is and features that they have. Can mean many things but common criteria to identify. Also operations for it like turning

this is a class. Defined by 2 main things: data it is going to contain and methods it is going to perform.

Need 3 elements to define class: 1. Specify what the properties are of an instance of this particular type/class. 2. Specify what the operations are. 3. Specify name of class we r creating

When want to use e.g. create an instance, have to provide acc values for the dif characteristics of object

Diagram

Description automatically generated

Coding examples of abstraction:

class Customer: # class always capitalised

# everything else always indented under

pass #keyword in python to indicate we r not doing anything

^ this is enough to satisfy python syntax.

Create new instance of a class:

def \_\_init\_\_(self, fname, lname) #self refers to current object and means the object we r currently creating

self.firstname = fname

self.lastname = lname

def print(self):

print(f’Full name: {self.fullName} {self.lastName}

Now in main .py can:

import Customer

cust = customer.Customer(‘Jyoti’, ‘Suresh’) #creating customer object

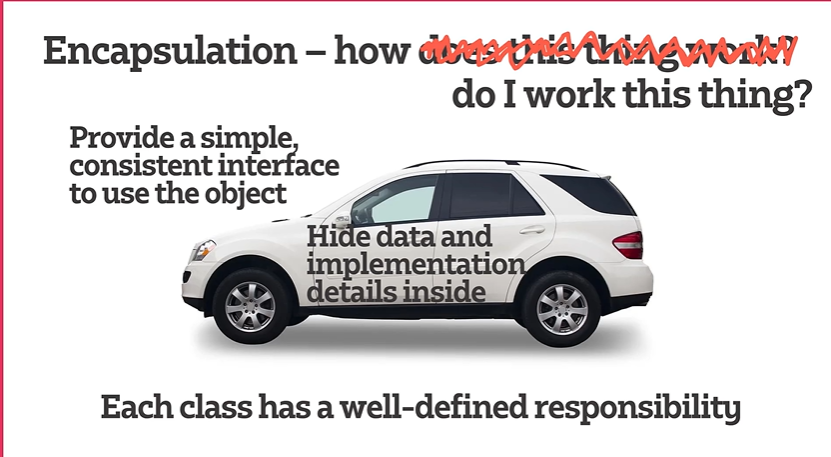
cust.print()

^ there are alt ways of initialising object

Encapsulation:

How does this work?

When u get into car, don’t need to know everything to work it, just need to know how to use:



Coding example:

No control over access to instance variables to class or any objects of that class without this. To stop things from changing, can use decorator e.g. @property

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Inheritance:

Relationship btwn classes

Graphical user interface

Description automatically generated

Aka base class and derived class

Inheritance hierarchy

Coding example:

Can pull up into superclass

Graphical user interface, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Polymorphism:

Will have an object and can tell it to do smth by sending it a msg. tell thing to do its job and it will know how to do it.

Graphical user interface, application

Description automatically generated

Coding example:

A screenshot of a computer

Description automatically generated

^ the print functions r polymorphism

Text

Description automatically generated

^another example

1. What are the main data types in python?

Main data types in python: int, float, str, bool, list, tuple, dict, set.

1. What is a variable?

A variable is a named container for storing values.

1. What is an if statement?

An if statement is a control structure that checks if a condition is met, and executes certain code if it is.

1. What is a for loop?

A for loop is a control structure that iterates over a collection, executing code for each item in the collection.

1. What is a while loop?

A while loop is a control structure that repeatedly executes code as long as a certain condition is met.

1. What is a function? Why are they useful?

A function is a named block of code that performs a specific task. They are useful for organization, code reuse and abstraction.

1. What are some good practices when writing functions?

Good practices when writing functions: using descriptive names, using docstrings, separating I/O, keeping functions short and simple.

1. What are the benefits of using classes? Are there disadvantages?

Benefits of using classes: encapsulation, abstraction, inheritance, and polymorphism. Disadvantages: harder to understand, may increase complexity.

1. What is instantiation?

Instantiation is the process of creating an instance of a class.

1. What is initialisation?

Initialization is the process of initializing the attributes of an instance of a class.

1. Is it a good idea for Data Engineers to use classes?

Yes, using classes can be useful for Data Engineers to organize and structure their code.

1. What is OOP?

OOP (Object-Oriented Programming) is a programming paradigm that organizes code around objects and classes, rather than procedures.

1. What are the 4 pillars of OOP?

The 4 pillars of OOP: encapsulation, abstraction, inheritance, and polymorphism.

1. How can OOP be useful for Data Engineers?

OOP can be useful for Data Engineers by allowing them to encapsulate data processing logic, encapsulate state and behaviour of data processing objects, reuse code and extend existing code.

1. What is a lambda function?

A lambda function is an anonymous, one-line function in Python.

1. What is unit testing?

Unit testing is a software testing technique where individual units of code are tested in isolation.

1. What are the advantages of unit testing?

Advantages of unit testing: early detection of errors, easier to maintain and modify code, supports refactoring, and increases confidence in code.

1. What is Test Driven Development (TDD)?

Test Driven Development (TDD) is a software development process where tests are written first, then code is written to make the tests pass.

1. Is unit testing useful for Data Engineers? Why?

Unit testing can be useful for Data Engineers because it can help them catch errors early, maintain code quality and make it easier to modify code in the future.

1. In what way can we manage errors in python?

Python provides built-in exceptions and error handling using try-except blocks. Additionally, using assertions and logging can also help manage errors.