CLASS, OBJ: OOPS PYTHON

Class, object, Variables inside a class, fun inside a class.

Accessing class Variable.

class <classname>:

Var = classname():
This is a object.

Similarly for functions.
Functions ⇒ def

Self.

class laptop:
price = 0
ram = " "

hp = laptop()
hp.price = 10,000
hp.ram = "8GB"

class is a template for the obj
OBJ ≡ INSTANCE of an class.

* Var inside class ⇒ class Variables & members of
* Func inside class ⇒ methods the class.

Creating obj, class instantiation.

__init__(self): CONSTRUCTOR:

It is a constructor. } INSTANCE/OBJ VARIABLE.
calls by default.
initialisation.

self → current object.
Calling by default

NOTE:
Inside a function always use self.var., coz function's 1st parameter is self and with the obj.

Each obj for those have self. have unique different Value.
They are NOT the same.
Hence they are called INSTACE VARIABLE.

To have a common Variable for a class, we call that CLASS VARIABLE.

We can change them ⇒ class.classVar = " "

TYPES OF METHODS:

INSTANCE METHOD ⇒ we use instance ≡ object, self.
Each obj. is accessed & used.

CLASS METHOD ⇒ accessing CLASS VARIABLE.

TYPES OF VAR:

Decorators are used above function.

STATIC METHOD: not using both class, instance variable.
@ classmethod
@ staticmethod.

In classmethod there should be parameter like self, : cls which takes class as argument.

If @ classmethod is NOT used then,
Class.classmethod(class)
If NOT
class classmethod()

INHERITANCE:

Single inheritance. — ① classnam(classname, -):
Multiple inheritance — 1 or more FROM CLASS accessing Via TO CLASS accessing DATA FROM HERE!

Inheritance is a concept when one class can access the member of other class.
Eg: son, mom, dad.

MULTI-LEVEL INHERITANCE (~~HOBBLE~~)
Inheriting in next next level.

HIERARCHIAL INHERITANCE:
if one class is being inherited by multiple class
Eg: dad, sons

Combination of inheritance is HYBRID INHERITANCE.

SUPER KEYWORD:
During inheritance if we have constructor for parent, child class.
child will be executed
Parent class will be executed, if there is no child class.

child class = derived class.

super() used to call a method from a parent
When we use multiple inheritance then the 1st parameter will be executed.

POLYMORPHISM

A program that changes it question according to the type of function call.

BASE ≡ PARENT CLASS
DERIVED ≡ CHILD CLASS.

Same function (different signature)
KEY DIFF: data type, no of arguments used.

There are 2 type of polymorphism

Static → Compile-time polymorphism
dynamic → Run-time polymorphism.
↓
Not Inheritance

inbuilt ⇒ len()
userdefined ⇒ default value

METHOD OVERRIDING.
A child provides a method that is already defined in its parent.

Static polymorphism is NOT accepted in Py.

ENCAPSULATION

Public & Private members.
Default public.
-- If Var name makes the Var private such that it can be accessed only within the class and CAN'T be accessed outside the class. } Private

This process is called encapsulation.
Hiding data, restricting data

_Varname } PROTECTED VARIABLE. (It just tells it should be handled with care).

Python uses name mangling. It redefines as _class_Var

EXCEPTION HANDLING:

3 types of error:

→ COMPILE TIME ERROR ↙ during compilation, outer syntax.

Each error has a name.

→ LOGICAL ERROR or logic error

→ RUN-TIME ERROR.

↳ during running of the program.

we should handle the errors.