Functions: A function is a set of code that performs some task. we can have some bundle of instructions kept together in a function by name. we can use this function at any time or anywhere.

- We have to use "return" Instead of print.
- return will call the function in function line

oops: (object-oriented programm)

object: Every Instance In Python are objects:

classes: classes are blueprint for similar objects

- the members of class are defined by the constructor __init__
- once we create constructor self parameter appears as it refers to the object that we are passing to the constructor

Inheritance: It allows us to define a class that inherits all the methods and properties from another class. Parent class is the class being inherited from, also called as base class. child class is the class the inherits from another class.

1) Single-level-Inheritance:

i) Parent class [F_1, F_2]
ii) child class [F_3, F_4]

- child class (parent class)
- child class [F_1, F_2, F_3, F_4]
- It allows a derivate class to inherit properties of only one parent class

2) Multi-level-Inheritance:

i) Parent class [F_1, F_2]
ii) child class [F_3, F_4]
iii) Grand child class [F_5]

- Grandchild class (child class)
- Grandchild class [F_1, F_2, F_3, F_4, F_5]
- features of Base class and derived class are inherited into new derived class.

3) Multiple Inheritance :

i) class (a) [1, 2]

ii) class (b) [3, 4]

iii) class (c) [5]

- class (c) (a, b)
- class (1, 2, 3, 4, 5)

when a class is derived from more than one base class it is called as multiple inheritance. the derived class inherits all the features of the base classes.

Encapsulation : It is a mechanism of wrapping the data and code acting on the data together as a unit single. In encapsulation the variables of class will be hidden from other classes and can only be accessed through the methods of their curent class.

Scope of variables : public Protected Private

Public variables can be used anywhere in the program. within func or outside func

Private variable are restricted within the function only.

- Private variables cannot be used outside the function.
- By default all variables are public

Public = a = [1,2,3]

Private = __a = [1,2,3]

Protected = _a = [1,2,3]

- Getters: this helps to access the private attributes from a class
- setters: helps to set the values to the private attributes in a class

Abstraction: Abstract class has atleast one abstract method in the class. the abstract method is nothing but it does not have any definition it only has a declaration and if you want to use them in python, since python by default doesnot support abstraction you need to install import module called as "ABC" (Abstract base classes)

Polymorphism: The same function being used for different types according to the situation

1) overloading.

2) overriding.

Overloading:

i) operator overloading:

Eg: "+" <-> concatination
<-> Addition

Here, we have "+" which we can use multiple ways. if we use it for string it is concatination if we use if for Integers we say it addition.

2) Method overloading:

Eg: add ( )
add (1,2)
add (1,2,3)

we can use default arguments in multiple ways.

overriding:

method overriding is a feature where the sub-class (child) can provide the programe withe specific implementation Proces of dat that are already defined in super-class (parent).

Transport( ) Father → cycle
↓
son — bike

obj = Transport ( )
obj. cycle ( ) & obj. bike ( )