**Classes & Objects in Python**

**Class:**

A class is a user-defined blueprint or prototype from which objects are created. Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by their class) for modifying their state.

The class creates a user-defined data structure, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object.

Some points on Python class:

* Classes are created by keyword class.
* Attributes are the variables that belong to a class.
* Attributes are always public and can be accessed using the dot (.) operator. Eg.: My class.Myattribute

Creating a Python Class:

Here, the class keyword indicates that you are creating a class followed by the name of the class (Dog in this case).

class Dog:

sound = "bark"

**Object:**

An Object is an instance of a Class. A class is like a blueprint while an instance is a copy of the class with actual values. It’s not an idea anymore, it’s an actual dog, like a dog of breed pug who’s seven years old. You can have many dogs to create many different instances, but without the class as a guide, you would be lost, not knowing what information is required.

An object consists of:

* State: It is represented by the attributes of an object. It also reflects the properties of an object.
* Behavior: It is represented by the methods of an object. It also reflects the response of an object to other objects.
* Identity: It gives a unique name to an object and enables one object to interact with other objects.

Declaring Class Objects (Also called instantiating a class):

When an object of a class is created, the class is said to be instantiated. All the instances share the attributes and the behavior of the class. But the values of those attributes, i.e. the state are unique for each object. A single class may have any number of instances.

**Example of Python Class and object:**

class Dog:

attr1 = "mammal"

attr2 = "dog"

def fun(self):

print("I'm a", self.attr1)

print("I'm a", self.attr2)

Rodger = Dog()

print(Rodger.attr1)

Rodger.fun()

**Output:**

mammal

I'm a mammal

I'm a dog