**Object Oriented programming:**

**Basic Terminologies:**

**Object:**

**It’s the instance of a class/ it’s the working entity of a class.**

**Class:**

**It is a template or blue print about the capability of what an object can do.**

**Method:**

**The behaviours of a class. It tells what a method can do. Comes after. Eg: .split**

**Instance:**

**Object and Instance both are same with small difference.**

#object-oriented programming

Method= function associated with class

Instance variable: contains unique attribute associated with class instance

Constructor: --init— python class relies on constructor to initialize (assign variables) any instance variables that the object will ness when it starts.

\_\_call\_\_method calls the initializer..

**Class variable**: can be accessed by every instances within the class

It can be changed for each instances.

Should be called using initializer (self) or class

Called using the class usually when the variable is constant through out

**Methods in class**

**Regular**: it takes self(initializer ) as default

**Class method**: takes class as the first argument.

@classmethod

Can be used as

**Static method:** it does not take either instances or class;

@staticmethod

**Inheritance-creating subclasses:**

issubclass(emp\_1), isinstance()….. is a built in function that checks the instances and subclasses.

**Special method:**

Allows to modify how the objects are printed and displayed

\_\_repr\_\_

\_\_str\_\_

\_\_add\_\_(dunder add)

**Property Decorators**—getters, setters, and deleters

@property lets the python use the methods getters, setters and deleters

The main benefit is that it lets the ones using our classes not go through the hassle of changing the code.

**SQLite:**

**Add, update, delete, select**

Library used to connect database with our program

=sqlite3.connect(‘employee.db’)

.execute lets us run the queries into the database

.commit is comiting to the action

.close is a good practice, as it closes the query

#COMMANDS

(“””CREATE TABLE EMPLOYEES()”””)

(“INSERT INTO ----- VALUES()”)

.execute("SELECT \* FROM employees WHERE last= 'SCHAFER'")

**To fetch from the table**

Fetchone, fetchoneall(), fetchmany(count)

While string formatting {} as a place holder can show sql injection error. Thus, the safe place holder to use is ?... ? benefit is we not need to specify the data type

**Decorators (allows to modify the function)**

It is function which takes another function as argument, adds desired functionality without changing the source code

It allows to manipulate the function with out changing the code in the function,

Function can be called from a function; function can have inner function.