**Question#1: Define object oriented programming language.**

**OOP:**

One of the popular approaches to solve a programming problem is by creating objects which gives the structures to program. This is known as Object-Oriented Programming (OOP).

**Question#2:** **List down the benefits of OOP.**

**BENEFITS OF OOP:**

* It is structure programming which makes the program simple to understand.
* Defined objects can be reuse in other programs; reusability is a benefit of OOP.
* Changes can easily take place, OOP can easily modify.
* Objects can maintain separately, making locating and fixing problems easier
* The code can be secured by using the abstraction mechanism.

**Question#3:** **Differentiate between function and method.**

## FUNCTION:

A function is a block of code to carry out a specific task, will contain its own scope and is called by name. All functions may contain zero (no) arguments or more than one arguments. On exit, a function can or cannot return one or more values.

**METHOD:**

A method in python is somewhat similar to a function, except it is associated with object/classes. Methods in python are very similar to functions except for two major differences.

* The method is implicitly used for an object for which it is called.
* The method is accessible to data that is contained within the class.

**Question#4:** **Define the following.**

## CLASS:

Class is the real world scenarios, and we create the different objects in the class. Class is the defining idea; classes can also have the sub classes which can inherit all or some characteristics of parent class.

1. **OBJECT:**

Objects are the instances of class, every object have the two characteristics i.e. attribute and behavior.

Example:

1. parrot is an object,

* Name, age, color are attributes.
* Singing, dancing is behavior.

1. **ATTRIBUTE:**

Attributes are the individual things that differentiate one object from another and determine the appearance, state, or other qualities of that object. Let's create a theoretical class called Motorcycle. A motorcycle class might include the following attributes and have these typical values:

Color: red, green, silver, brown

Style: cruiser, sport bike, standard

Make: Honda, BMW, Bultaco

1. **BEHAVIOR:**

Behavior is the only way objects can do anything to themselves or have anything done to them. For example, to go back to the theoretical Motorcycle class, here are some behaviors that the Motorcycle class might have:

* Start the engine
* Stop the engine
* Speed up
* Change gear
* Stall

**Question#5:** **Write a code in python in which creates a class named it Car which have 5 attributes such like (model, color and name etc.) and 3 Methods. And create 5 object instances from that class.**

<https://github.com/maryamikram/Python-Assignments/blob/master/ass%236%20Question%235.ipynb>

class Car():

def \_\_init\_\_(self, model, year, color, making, fuel):

self.model = model

self.year = year

self.color = color

self.making = making

self.fuel = fuel

car1 = Car('Cultus VXR', '2015', 'Silver', 'Suzuki', 'Petrol')

car2 = Car('Mira', '2012', 'White', 'Daihatsu', 'Petrol')

car3 = Car('Minica', '2013', 'White', 'Mitsubishi', 'Petrol')

car4 = Car('Cultus VLX', '2007', 'Blue', 'Suzuki', 'CNG')

car5 = Car('Cuore', '2005', 'Black', 'Daihatsu', 'CNG')

def info(c):

print(c.model)

print(c.year,'\n\n')

info(car4)

def A(a):

print("Model: ",a.model)

print("Color: ",a.color)

print("Year: ",a.year)

print("Fuel: ",a.fuel)

print("Making: ",a.making)

print("\n\n")

A(car2)

def car\_info(info):

print("Model: ",info.model)

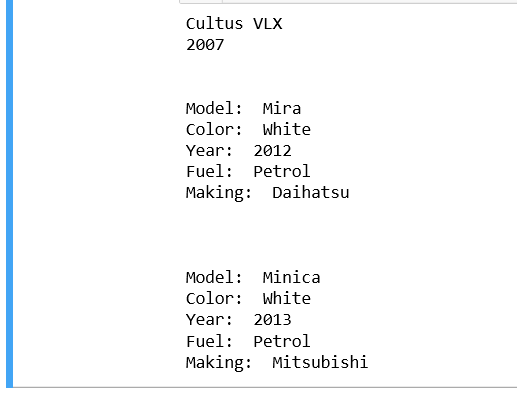
print("Color: ",info.color)

print("Year: ",info.year)

print("Fuel: ",info.fuel)

print("Making: ",info.making)

car\_info(car3)

**OUTPUT:**