**Question 1:**

**Define Object Oriented Programming Language?**

**Answer:**

Object-oriented programming is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields, and code, in the form of procedures. A feature of objects is an object's procedures that can access and often modify the data fields of the object with which they are associated.

**Question 2:**

**List down the Benefits of OOP?**

**Answer:**

* Modularity for easier troubleshooting. Something has gone wrong, and you have no idea where to look.
* Reuse of code through inheritance.
* Flexibility through polymorphism.
* Effective problem solving.

**Question 3:**

**Differentiate between function and method?**

**Answer:**

A **function** is a piece of code that is called by name. It can be passed data to operate on (i.e. **the** parameters) and can optionally return data (**the** return value). All data that is passed to a **function** is explicitly passed. A **method** is a piece of code that is called by a name that is associated with an object.

**Question 4:**

**Define the following terms:**

**1. Class**

**2. Object**

**3. Attribute**

**4. Behavior**

**Answer:**

**Class:** In object-oriented programming, a class is an extensible program-code-template for creating objects, providing initial values for state and implementations of behavior.

**Object:** In **object-oriented** programming (**OOP**), **objects** are the things you think about first in designing a program and they are also the units of code that are eventually derived from the process. Each **object** is an instance of a particular class or subclass with the class's own methods or procedures and data variables.

**Attribute:** In **Object-oriented** programming (**OOP**), classes and objects have **attributes**. **Attributes** are data stored inside a class or instance and represent the state or quality of the class or instance. One can think of **attributes** as noun or adjective, while methods are the verb of the class.

**Behavior:** A class's **behavior** determines how an instance of that class operates; for example, how it will "react" if asked to do something by another class or object or if its internal state changes. **Behavior** is the only way objects can do anything to themselves or have anything done to them.

**Question 5:**

**Write a code in python in which create a class named it Car which have 5 attributes such like (model, color and name etc.) and 3 methods And create 5 object instance from that class.**

**Answer:**

class Car:

car\_count = 0

def start(self, name, make, model):

print ("Engine started")

self.name = name

self.make = make

self.model = model

Car.car\_count += 1

car\_a = Car()

car\_a.start("Corrola", "Toyota", 2015)

print(car\_a.name)

print(car\_a.car\_count)

car\_b = Car()

car\_b.start("City", "Honda", 2013)

print(car\_b.name)

print(car\_b.car\_count)