Question 1:

Define Object Oriented Programming Language?

Ans. Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). 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 (objects have a notion of "this" or "self"). In OOP, computer programs are designed by making them out of objects that interact with one another. OOP languages are diverse, but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

Question 2:

List down the Benefits of OOP?

Ans.

1. It provides a clear modular structure for programs which makes it good for defining abstract data types in which implementation details are hidden
2. Objects can also be reused within an across applications. The reuse of software also lowers the cost of development. More effort is put into the object-oriented analysis and design, which lowers the overall cost of development.
3. It makes software easier to maintain. Since the design is modular, part of the system can be updated in case of issues without a need to make large-scale changes
4. Reuse also enables faster development. Object-oriented programming languages come with rich libraries of objects, and code developed during projects is also reusable in future projects.
5. It provides a good framework for code libraries where the supplied software components can be easily adapted and modified by the programmer. This is particularly useful for developing graphical user interfaces.
6. Better Productivity as OOP techniques enforce rules on a programmer that, in the long run, help her get more work done; finished programs work better, have more features and are easier to read and maintain.

Question 3:

Differentiate between function and method?

Ans. 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 whereas 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

Ans.

Class:

A class describes the contents of the objects that belong to it: it describes an aggregate of data fields (called instance variables), and defines the operations (called methods).

Object:

An object is an element (or instance) of a class; objects have the behaviors of their class. The object is the actual component of programs, while the class specifies how instances are created and how they behave.

Attributes:

The data stored inside a class or instance and represent the state or quality of the class or instance is Attribute.

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 it’s 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.

In [9]:

```
class Car:
    def __init__(self,company,model,color,name,enginepower):
        self.company = company
        self.model = model
        self.color = color
        self.name = name
        self.enginepower = enginepower

    def car_start(self):
        print("The car "+self.name+" is started")
    def car_run(self):
        print("The car "+self.name+" is running")
    def car_stop(self):
        print("The car "+self.name+" is stopped\n")

car1=Car("Suzuki",2000,"Blue","Mehran","700CC")
car1.car_start()
car1.car_run()
car1.car_stop()

car2=Car("Honda",2000,"White","Civic","1000CC")
car2.car_start()
car2.car_run()
car2.car_stop()

car3=Car("Toyota",2000,"Green","Corolla","500CC")
car3.car_start()
car3.car_run()
car3.car_stop()
```

```
car3=Car("Toyota",2000,"Green","Corolla","500CC")
car3.car_start()
car3.car_run()
car3.car_stop()

car4=Car("Hyundai",2000,"Black","Creta","1200CC")
car4.car_start()
car4.car_run()
car4.car_stop()

car5=Car("FAW",2000,"Red","Hawk","2000CC")
car5.car_start()
car5.car_run()
car5.car_stop()
```

```
The car Mehran is started
The car Mehran is running
The car Mehran is stopped

The car Civic is started
The car Civic is running
The car Civic is stopped

The car Corolla is started
The car Corolla is running
The car Corolla is stopped

The car Creta is started
The car Creta is running
The car Creta is stopped

The car Hawk is started
```