**ASSIGNMENT 6**

**Question 1:**

**Define Object Oriented Programming Language?**

**Answer:**

**Object**-**oriented programming** (**OOP**) refers to a type of computer **programming** (software design) in which programmers **define** not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure.

**Question 2:**

**List down the Benefits of OOP?**

**Answer:**

Some of the advantages of object-oriented programming include:

1. Improved software-development productivity: Object-oriented programming is modular, as it provides separation of duties in object-based program development. It is also extensible, as objects can be extended to include new attributes and behaviors. Objects can also be reused within an across applications
2. Improved software maintainability: For the reasons mentioned above, objectoriented software is also 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.
3. Faster development: Reuse enables faster development. Object-oriented programming languages come with rich libraries of objects, and code developed during projects is also reusable in future projects.
4. Lower cost of development: The reuse of software also lowers the cost of development. Typically, more effort is put into the object-oriented analysis and design, which lowers the overall cost of development.

**Question 3:**

**Differentiate between function and method?**

**Answer:**

Python function is a sequence of statements that execute in a certain order, given a name. They let us implement code reusability. When we talked about Python Functions, we talked about built-in and user-defined functions.

Python method is like a function, except it is attached to an**object**. We call a method on an object, and it possibly makes changes to that object. A method, then, belongs to a **class**

**Question 4:**

**Define the following terms:**

**1. Class**

**2. Object**

**3. Attribute**

**4. Behavior**

**Answer:**

**1. Class:**

A class is a blueprint for objects- one class for any number of objects of that type. You can also call it an abstract data type. Interestingly, it contains no values itself, but it is like a prototype for objects. To define a class in python programming, we use the ‘class’ keyword. This is like we use ‘def’ to define a **function in python.**

**2. Object:**

Object is simply a collection of data (variables) and methods (functions) that act on those data. And, class is a blueprint for the object.

**3. Attribute:**

An attribute is defined as a quality or characteristic of a person, place, or thing. Real life individuals and fictional characters possess various attributes. An Attribute in Python is a name belonging to an object - a method or a variable.

**4. Behavior:**

With Behavior, we are writing a specification for the piece of code we are about to write. It is much clearer with Behavior driven development that the specification is written before the code.

**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:**

**Repository is uploaded named assignment6python**

**CODE:**

class Car:

def \_\_init\_\_(self, name, model,color,enginecc,origin):

self.name = name

self.model = model

self.color = color

self.enginecc = enginecc

self.origin = origin

def myfunc(self):

print("Car Details\n" )

def myfunc2(self):

print("Owner and car model\n" + self.name+ self.model)

def myfunc3(self):

print("Car Specs and Info\n"+self.color+ self.enginecc +self.origin)

c1 = Car("John\n", "Ferrari\n","red\n","1800cc\n","Made in Japan")

c1.myfunc()

c2 = Car("NAME:John\n", "MODEL:Ferrari\n","red\n","1800cc\n","Made in Japan")

c2.myfunc2()

c3 = Car("John\n", "Ferrari\n","COLOR:Red \n","ENGINE:1800cc\n","ORIGIN:Made in Japan")

c3.myfunc3()