**Question No.1**

**Python OOPs Concepts**

Python is an object-oriented programming language. It allows us to develop applications using an Object Oriented approach. In Python, we can easily create and use classes and objects.

Major principles of object-oriented programming system are given below.

* Object
* Class
* Method
* Inheritance
* Polymorphism
* Data Abstraction
* Encapsulation

**Question No.2**

**Benefits of OOP:**

* OOP provides a clear modular structure for programs.
* It is good for defining abstract data types.
* Implementation details are hidden from other modules and other modules has a clearly defined interface.
* It is easy to maintain and modify existing code as new objects can be created with small differences to existing ones.
* objects, methods, instance, message passing, inheritance are some important properties provided by these particular languages
* encapsulation, polymorphism, abstraction are also counts in these fundamentals of programming language.
* It implements real life scenario.
* In OOP, programmer not only defines data types but also deals with operations applied for data structures.

**Question No. 3**

**Python Method**

* Method is called by its name, but it is associated to an object (dependent).
* A method is implicitly passed the object on which it is invoked.
* It may or may not return any data.
* A method can operate on the data (instance variables) that is contained by the corresponding class

**Functions**

* A function is a block of code that is also called by its name. (independent)
* The function can have different parameters or may not have any at all. If any data (parameters) are passed, they are passed explicitly.
* It may or may not return any data.
* Function does not deal with Class and its instance concept.

**Difference between method and function**

Simply, function and method both look similar as they perform in an almost a similar way, but the key difference is the concept of ‘Class and its Object‘.

Functions can be called only by its name, as it is defined independently. But methods can’t be called by its name only, we need to invoke the class by a reference of that class in which it is defined, i.e. method is defined within a class and hence they are dependent on that class

**Question No. 4**

**Class**

The class can be defined as a collection of objects. It is a logical entity that has some specific attributes and methods. For example: if we have an employee class then it should contain an attribute and method, i.e. an email id, name, age, salary, etc.

**Object**

Object is simply a collection of data (variables) and methods (functions) that act on those data.

**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. In short, attributes store information about the instance.

**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.

**Task # 5**

class Car:

def \_\_init\_\_(self, model, color, name,VIN,brand):

self.model = model

self.color = color

self.name = name

self.VIN = VIN

self.brand = brand