**Python Programming (Saylani) Assignment#6**

**Q#1: DEFINE OBJECT ORIENTED PROGRAMMING LANGUAGE?**

**Object-oriented programming:** As the name suggests, Object-Oriented Programming or OOPs refers to languages that uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

**Q#2: LIST DOWN THE BENEFITS OF OOP?**

**OOPs Concepts:**

* Polymorphism
* Inheritance
* Encapsulation
* Abstraction
* Class
* Object
* Method
* Message Passing

**Q#3:** **DIFFERENTIATE BETWEEN FUNCTION AND METHOD?**

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.

1. Method is called by its name, but it is associated to an object (dependent) whereas Function is block of code that is also called by its name. (independent)
2. A method is implicitly passed the object on which it is invoked whereas the function can have different parameters or may not have any at all. If any data (parameters) are passed, they are passed explicitly.
3. Both may or may not return any data.
4. A method can operate on the data (instance variables) that is contained by the corresponding class whereas Function does not deal with Class and its instance concept.

**Q#4: DEFINE THE FOLLOWING TERMS:**

**OBJECT:**

An object is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. Object determines the behavior of the class. When you send a message to an object, you are asking the object to invoke or execute one of its methods.

**CLASS:**

A class is a template that defines the form of an object. It specifies both the data and the code that will operate on that data.

**ATTRIBUTE:**

Attributes are the individual things that differentiate one object from another and determine the appearance, state, or other qualities of that object. Attributes of an object can also include information about its state;

**BEHAVIOR:**

A class's behavior determines how an instance of that class operates; To define an object's behavior, you create methods.

From a programming point of view, an object can be a data structure, a variable or a function. It has a memory location allocated.

**Q#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.**

**def makeCarClass():**

**def initialize(self, model, color, currSpeed, maxSpeed, mpg):**

**self["model"] = model**

**self["color"] = color**

**self["currSpeed"] = currSpeed**

**self["maxSpeed"] = maxSpeed**

**self["mpg"] = mpg**

**return self**

**def changeSpeed(self, amount):**

**newSpeed = self["currSpeed"] + amount**

**if newSpeed >= self["maxSpeed"]:**

**self["currSpeed"] = self["maxSpeed"]**

**elif newSpeed <=0:**

**self["currSpeed"] = 0**

**else:**

**self["currSpeed"] = newSpeed**

**return {"initialize":initialize, "changeSpeed":changeSpeed, "move":move}**

**car1 = carClass["initialize"]({},"Ferrari", "white", 0, 200, 10, 40, 0)**

**car2 = carClass["initialize"]({},"Honda", "black", 0, 120, 30, 40, 0)**

**print(car1,car2)**