**ASSIGNMENT # 06**

**Q#1:**

Object-oriented programming (OOP):

Object-oriented programming (OOP) is a programming language model in which programs are organized around data, or objects, an object can be defined as a data field that has unique attributes and behavior.

Once an object is known, it is generalized as a class of objects that defines the kind of data it contains and any logic sequences that can manipulate it.

There are 4 major principles that make a language Object Oriented. These are Encapsulation, Data Abstraction, Polymorphism and Inheritance. These are also called as four pillars of Object-Oriented Programming.

**Q#2:**

Benefits of OOP:

* User can create new data type or users define data type by making class.
* Code can be reuse by using inheritance.
* Data can be hiding from outside world by using encapsulation.
* Operators or functions can be overloaded by using polymorphism, so same functions or operators can be used for multitasking.

**Q#3:**

Function:

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.

Method:

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.

**Q#4:**

Class:

A **class** is a code template for creating objects. Objects have member variables and have behaviour associated with them. In **python** a **class** is created by the keyword **class** . An object is created using the constructor of the **class.** This object will then be called the instance of the class.

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. In between, each object is made into a generic class of object and even more generic classes are defined so that objects can share models and reuse the class definitions in their code. Each object is an instance of a particular class or subclass with the class's own methods or procedures and data variables.

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. For example, someone might be labeled beautiful, charming, funny, or intelligent.

Behavior:

Behavior provides a language suitable for Behavior Driven Development (BDD), as opposed to post-coding unit testing. It operates as a facade over Python’s unit testing framework.

**Q#5:**

class car():

wheels = 4

def \_\_init\_\_(self, model, color, name, brand, transmission\_type, speed ):

self.model = model

self.color = color

self. name = name

self.brand = brand

self.transmission\_type = transmission\_type

self.speed = 0

def getSpeed(self):

return self.speed

def accelerate(self):

self.speed += 10

def brake(self):

self.speed -= 10

print (car.wheels)

print ("-------object-1------- ")

obj1 = car(2013,"black","city","Honda","Manual", 32)

print("The current speed of car 1 is: ", obj1.getSpeed() )

obj1.accelerate()

print("The current speed of car 1 is: ", obj1.getSpeed() )

obj1.accelerate()

print("The current speed of car 1 is: ", obj1.getSpeed() )

obj1.accelerate()

print("The current speed of car 1 is: ", obj1.getSpeed() )

obj1.brake()

print("The current speed of car 1 is: ", obj1.getSpeed() )

print ("-------object-2------- ")

obj2 = car(2019,"black","civic","Honda","Automatic", 32)

print("The current speed of car 1 is: ", obj2.getSpeed() )

obj2.accelerate()

print("The current speed of car 1 is: ", obj2.getSpeed() )

obj2.accelerate()

print("The current speed of car 1 is: ", obj2.getSpeed() )

obj2.accelerate()

print("The current speed of car 1 is: ", obj2.getSpeed() )

obj2.brake()

print("The current speed of car 1 is: ", obj2.getSpeed() )

print ("-------object-3------- ")

obj3 = car(2003,"beige","cultus","Suzuki","Manual", 12)

print("The current speed of car 1 is: ", obj3.getSpeed() )

obj3.accelerate()

print("The current speed of car 1 is: ", obj3.getSpeed() )

obj3.accelerate()

print("The current speed of car 1 is: ", obj3.getSpeed() )

obj3.accelerate()

print("The current speed of car 1 is: ", obj3.getSpeed() )

obj3.brake()

print("The current speed of car 1 is: ", obj3.getSpeed() )

print ("-------object-4------- ")

obj4 = car(2015,"black","Mehran","Suzuki","Manual", 20)

print("The current speed of car 1 is: ", obj4.getSpeed() )

obj4.accelerate()

print("The current speed of car 1 is: ", obj4.getSpeed() )

obj4.accelerate()

print("The current speed of car 1 is: ", obj4.getSpeed() )

obj4.accelerate()

print("The current speed of car 1 is: ", obj4.getSpeed() )

obj4.brake()

print("The current speed of car 1 is: ", obj4.getSpeed() )

print ("-------object-5------- ")

obj5 = car(2010,"White","Corolla","Toyota","automatic", 2)

print("The current speed of car 1 is: ", obj5.getSpeed() )

obj5.accelerate()

print("The current speed of car 1 is: ", obj5.getSpeed() )

obj5.accelerate()

print("The current speed of car 1 is: ", obj5.getSpeed() )

obj5.accelerate()

print("The current speed of car 1 is: ", obj5.getSpeed() )

obj5.brake()

print("The current speed of car 1 is: ", obj5.getSpeed() )



