Assignment # 6

Question 1: Define Object Oriented Programming Language?

Answer: 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.[1][2] OOP languages are diverse, but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

Many of the most widely used programming languages (such as C++, Java, Python, etc.) are multi-paradigm and they support object-oriented programming to a greater or lesser degree, typically in combination with imperative, procedural programming. Significant object-oriented languages include Java, C++, C#, Python, PHP, JavaScript, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Common Lisp, MATLAB, and Smalltalk.

Question 2: List down the Benefits of OOP?

Answer: Some of the advantages of OOP are mentioned below:

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

Question 3: Differentiate between function and method?

Answer: 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. In most respects it is identical to a function except for two key differences:

1. A method is implicitly passed the object on which it was called.
2. A method is able to operate on data that is contained within the class (remembering that an object is an instance of a class - the class is the definition, the object is an instance of that data).

Question 4: Define the following terms:

1. 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.

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 : A class attribute is a Python variable that belongs to a class rather than a particular object. It is shared between all the objects of this class and it is defined outside the constructor function, \_\_init\_\_(self,...) , of the class.

4. Behavior : They are used to define the behaviors(uses/functionality) of an object. Methods are functions defined inside the body of a class. They tell what function do the class contain and what powers do it have, and what kind of task it can perform.

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.

class Car:

def \_\_init\_\_(self,model,color,name,cc,speakers):

self.model = model

self.color = color

self.name = name

self.cc = cc

self.speakers = speakers

def printDetails(self):

print("Car Name : {}\nCar Color : {}\nCar Model : {}\nCC : {}\nSpeakers? : {}".format(self.name,self.color,self.model,self.cc,self.speakers))

print("---------------------------------------------------------------------------")

def updateSpeakers(self,updatedData):

self.speakers = updatedData

def updateColor(self,updatedColorData):

self.color = updatedColorData

myObjectOne = Car("2019","Black","Honda","1600cc","Yes")

myObjectOne.printDetails()

myObjectOne.updateSpeakers("No")

myObjectOne.updateColor("Metallic Black")

print("Data Updated")

myObjectOne.printDetails()

myObjectTwo = Car("2018","Blue","Audi","3200cc","Yes")

myObjectTwo.printDetails()

myObjectTwo.updateSpeakers("No")

myObjectTwo.updateColor("Red")

print("Data Updated")

myObjectTwo.printDetails()

myObjectThree = Car("2017","Grey","Toyota","1800cc","Yes")

myObjectThree.printDetails()

myObjectThree.updateSpeakers("No")

myObjectThree.updateColor("White")

print("Data Updated")

myObjectThree.printDetails()

myObjectFour = Car("2016","White","Mazda","900cc","No")

myObjectFour.printDetails()

myObjectFour.updateSpeakers("Yes")

myObjectFour.updateColor("Brown")

print("Data Updated")

myObjectFour.printDetails()

myObjectFive = Car("2015","Brown","Civic","1700cc","Yes")

myObjectFive.printDetails()

myObjectFive.updateSpeakers("No")

myObjectFive.updateColor("White")

print("Data Updated")

myObjectFive.printDetails()