**ASSIGMENT NO 06 PAYTHON**

1. **Define objected oriented Programming language**

An “Object” is the key component of Object-oriented programming. An Object may contain data (fields or variables) or code (methods or procedures). The creation of these objects is based on a programmer-defined blue-print also known as a Class.

1. **List down the benefit of OOP**

* Modularity for easier troubleshooting. Something has gone wrong, and you have no idea where to look.
* Reuse of code through inheritance.
* Flexibility through polymorphism.
* Effective problem solving.

1. **Difference between function and method**

* **F**unction → **F**ree (Free means not belong to an object or class)
* **M**ethod → **M**ember (A member of an object or class)

Method and a function are the same, with different terms. A method is a procedure or function in object-oriented programming.

A function is a group of reusable code which can be called anywhere in your program. This eliminates the need for writing the same code again and again. It helps programmers in writing modular codes.

1. Define following term

**Class**

A class is an entity that determines how an object will behave and what the object will contain. In other words, it is a blueprint or a set of instruction to build a specific type of object.

**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.

**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

Q no 05

class Vehicle(object): # no instance of this class should be created

def \_\_init\_\_(self, typ, make, model, color, year, miles):

self.typ = typ

self.make = make

self.model = model

self.color = color.lower()

self.year = year

self.miles = miles

def vehicle\_print(self):

print('Vehicle Type: ' + str(self.typ))

print('Make: ' + str(self.make))

print('Model: ' + str(self.model))

print('Color: ' + str(self.color))

print('Year: ' + str(self.year))

print('Miles driven: ' + str(self.miles))