## *O bject Oriented Programming Language.*

***D efine Object Oriented Programming Language?***

Object-oriented programming (OOP) refers to a type of computer programming (software )operations (functions) that can be applied to the data structure.

In this way, the data structure becomes an object that includes both data and functions. In addition,

programmers can create relationships between one object and another. For example, objects can inherit characteristics from other objects.

## *P ython OOPs Concepts*

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

## *L ist down the Benefits of OOP?*

The advantages of OOP are mentioned below:

# 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.

## *D ifferentiate between function and method?*

|  |  |
| --- | --- |
| ***M ethod*** | ***F unctions*** |
| 1. Method is called by its name, but it is associated to an object (dependent). 2. A method is   implicitly passed the object on which it is invoked. | 1. Function is block of code that is also called by its name. (independent) 2. The function can have different parameters or may not have any at all. If any data |

* + 1. It may or may not return any data.
    2. A method can operate on the data (instance variables) that is contained by the corresponding class

(parameters) are passed, they are passed explicitly.

1. It may or may not return any data.
2. Function does not deal with Class and its instance concept.

## *D efine the following terms:*

1. ***. 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

## *. Attribute*

A class attribute is a variable that belongs to a certain class, and not a particular object. Every instance of this class shares the *same* variable. These attributes are usually defined outside the init constructor.

1. ***. Behavior*** Objects in Python are generally classified

according to their behaviors and the features that they

implement. For example, all of the sequence types such as strings, lists, and tuples are grouped together