# Characteristics of Class

- Class contains two things as Characteristics and Behaviours.
- Characteristics are considered as a variables which are part of class.
- Object-oriented programming allows for variables to be used at the class level or the instance level.
- Variables are essentially symbols that stand in for a value you're using in a program.
- At the class level, variables are referred to as class variables, whereas variables at the instance level are called instance variables.
- When we expect variables are going to be consistent across instances, or when we would like to initialize a variable, we can define that variable at the class level.
- When we anticipate the variables will change significantly across instances, we can define them at the instance level.

In python there are two types of variables as
1. Instance variable
2. Class variable

**Class Variable :**
- Class variables are defined within the class construction.
- Because they are owned by the class itself, class variables are shared by all instances of the class.
- They therefore will generally have the same value for every instance unless you are using the class variable to initialize a variable.
- Defined outside of all the methods, class variables are, by convention, typically placed right below the class header and before the constructor method (init) and other methods.
- We can access class variables without creating the object of a class by using class name.

**Instance Variables :**
- Instance variables are owned by instances of the class.
- This means that for each object or instance of a class, the instance variables are different.
- Unlike class variables, instance variables are defined within init method.
- We can access instance variable after creating object.

## Consider below application which demonstrates types of characteristics

```
print("---- Marvellous Infosystems by Piyush Khairnar-----")

print("Demonstration of Characteristics of Class")

class Demo:
    x = 10

    def __init__(self,no1,no2):
        self.i = no1
        self.j = no2
```

```
obj1 = Demo(10,20)
obj2 = Demo(11,21)

print(obj1.i)
print(obj1.j)

print(obj2.i)
print(obj2.j)

print(Demo.x)
```

In above application there is one class named as Demo. That class contains X as a class variable and i,j as a instance variables.

## Output of above application