# Object Oriented Programming Language:

#### Object-oriented programming is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields, and code, in the form of procedures.

# Benefits of OOP:

#### 1. Re-usability

#### 2. Data Redundancy

#### 3. Code Maintenance

#### 4. Security

#### 5. Design Benefits

#### 6. Better productivity

#### 7. Easy troubleshooting

Differentiate between function and method

|  |  |
| --- | --- |
| **Function** | **Method** |
| * Functions have independent existence means they can be defined outside of the class. | * Methods do not have independent existence they are always defined with in class. |
| * Functions are defined in structured languages | * Methods are defined in object oriented languages |
| * Functions do not have any reference variable. | * Methods are called using reference variable. |
| .   * Functions are self-describing unit of code. | * Methods are used to manipulate instance variable of a class. |

Definitions:

# Class:

A class is a code template for creating objects. Objects have member variables and have behavior associated with them. In python a class is created by the keyword class.

# Object:

An instance of a class. This is the realized version of the class, where the class is manifested in the program.

# Attributes:

## There are two types of attributes in python

* An instance attribute is a Python variable belonging to one, and only one, object. This variable is only accessible in the scope of this object and it is defined inside the constructor function, \_\_init\_\_(self,..) of the class.
* 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.

# 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 merely because they all happen to support a common set of sequence operations such as *s*[*n*], len(*s*), etc.