Classes and Objects

Basic Introduction to OOP

- Python is an procedural and Object Oriented Language.
- Basic Terminologies in OOPS are:

➢**Class:** -It is an user defined prototype for an object that defines a set of attributes that characterize any object of the class.

➢**Class variable:-** A variable that is shared by all instances of class and is defined within a class and outside of any class method.

➢**Data member:-**A class variable or instance variable that holds data associated with a class and its objects.

➢**Function overloading :-** Using same name of function to perform various operation.

➢**Instance variable: -** A variable that is defined inside a method and belongs only to a current instance of a class.

Basic Introduction to OOP

- **Instance: -** An individual Object of a certain class.
- **Instantiation: -** Process of creation of an instance.
- **Method: -** Special type of function that is defined in a class definition.
- **Object: -** A unique instance of a class.
- **Operator Overloading : -** Using same name of function to perform various operation.
- **Inheritance : -**Process of deriving the sub class from the super class
- **Polymorphism:-**One name multiple functionalities.
- **Encapsulation:-**Binding of data and members together.

How to Define a class

- It uses a keyword called as **class** followed by the name of the class.

Syntax:-

```
class class_name:
  Body of the class
```

We can use documentation string in the class and can access it whenever necessary by usinsg classname.__doc__

Accessing Attributes

You can add, remove, or modify attributes of classes and objects at any time .

- The **getattr(obj, name[, default])** – to access the attribute of object.
- The **hasattr(obj,name)** – to check if an attribute exists or not.
- The **setattr(obj,name,value)** – to set an attribute. If attribute does not exist, then it would be created.
- The **delattr(obj, name)** – to delete an attribute

Built-In Class Attributes

Every Python class keeps following built-in attributes and they can be accessed using dot operator like any other attribute −

- **__dict__** − Dictionary containing the class's namespace.
- **__doc__** − Class documentation string or none, if undefined.
- **__name__** − Class name.
- **__module__** − Module name in which the class is defined. This attribute is "__main__" in interactive mode.
- **__bases__** − A possibly empty tuple containing the base classes, in the order of their occurrence in the base class list.

Task

1. Define a class for vehicles. Create two new vehicles called car1 and car2. Set car1 to be a red convertible worth $60,000.00 with a name of Fer, and car2 to be a blue van named Jump worth $10,000.00. Print out the car’s description.