1. **A class is used to define a blueprint for a given object, whereas a module is used to reuse a given piece of code inside another program**.
2. To create instances of a class, you **call the class using class name and pass in whatever arguments its \_\_init\_\_ method accepts**.
3. Class attributes are attributes which are owned by the class itself. They will be shared by all the instances of the class. Therefore they have the same value for every instance. We define class attributes outside all the methods, **usually they are placed at the top, right below the class header**.
4. Instance attributes are **defined in the constructor**. Defined directly inside a class. Defined inside a constructor using the self parameter.
5. The self is used to represent the [instance](https://www.edureka.co/blog/isinstance-in-python/) of the class. With this keyword, you can access the attributes and methods of the [class in python](https://www.edureka.co/blog/python-class/). It binds the attributes with the given arguments.
6. When we use an operator on user defined data types then automatically a special function or magic function associated with that operator is invoked. Changing the behavior of operator is as simple as changing the behavior of method or function. You define methods in your class and operators work according to that behavior defined in methods. When we use + operator, the magic method \_\_add\_\_ is automatically invoked in which the operation for + operator is defined. There by changing this magic method’s code, we can give extra meaning to the + operator.
7. Operator Overloading means giving extended meaning beyond their predefined operational meaning. For example operator + is **used to add two integers as well as join two strings and merge two lists**. It is achievable because '+' operator is overloaded by int class and str class.
8. A very popular and convenient example is the **Addition (+) operator**.
9. In order to develop robust and well-designed software products with Python, it is essential to obtain a comprehensive understanding of OOP. In this article, we will elaborate on two key concepts of OOP which are **inheritance and polymorphism**