25-07-24

Objective: Documentation of OOPs concept in python

**OOPS IN PYTHON**

**Class and object**

Class is the blueprint of the object.

Object is the instance of class

Implemented a basic code to understand class, object and init method.

* Created a class called **Event**. Used **\_\_init\_\_** method which works like a constructor .It gets executed when the class is being initiated.
* Self basically refers to the instance of the class that is being created. Acts like a pointer.
* Created another method called **display** to display the event name.
* Created an object called obj and called the class with the self.eventname will fetch the name.
* And display it by the help of object.

**Four concepts of OOPS**

1. Inheritance

2. Encapsulation

3. Abstraction

4. Polymorphism

**Inheritance** allows us to inherit properties and methods from one class to another class. **Parent class** is called as the **base class**. **Child class** is known as the **derived class**.

* Created a class named **P**
* Used the **\_\_init\_\_** method takes argument for the first name and the last name.
* Took another function called **print name** to print the first name and the last name.
* Created another class **S** which is inherited from P, which means P is the base class and S is the child class or the derived class.
* Created an object called x and calling the function S,the derived class which does not have any functions in it but inherits from the parent class.
* Lastly calling the print name through the object created.

**Encapsulation** is binding or wrapping up of data into single unit.

Here we create a class called Employee.

Create a \_\_init\_\_ method which takes attributes of name, salary and project.

Creating two methods display and set.Where in display we only print name and salary through self.

And using set for printing project.

Creating an object called emp and passing data that we want.

Emp.show and emp.work will call both the methods.

In encapsulation, we use mangling.

Mangling is a technique used to make private attributes.We can access private attributes by using it with object class and attribute name 🡪 emp.\_Employee\_\_project.

**Abstraction** is a fundamental concept of hiding unnecessary information and exposing only the important details.

In python, abstraction can be achieved by using the two methods that are abstract class and interface.

A class that consists of one or more than one abstract method are called abstract classes. Abstract classes cannot be instantiated, which means we cannot create an object.

**Polymorphism** means the same function name being used for different types.

Created a class called car.

Created two functions or methods named intro and speed.

Created another two classes named as Honda and Hyundai which are inherited from the base class car.

Used the same function speed in both this classes.

Created three objects from the three classes.

Called all the three classes with both the functions.

The method speed is overriden in both the classes below.

The intro method is inherited from the class car.

The speed method overrides both classes Honda and Hyundai.

Sincerely,

Sambodhi Kotekar

Sambodhi.kotekar@inteliment.com