INTRODUCTION TO OBJECT ORIENTED PROGRAMMING (OOP) USING PYTHON

Vishwa Mittar - 18CS

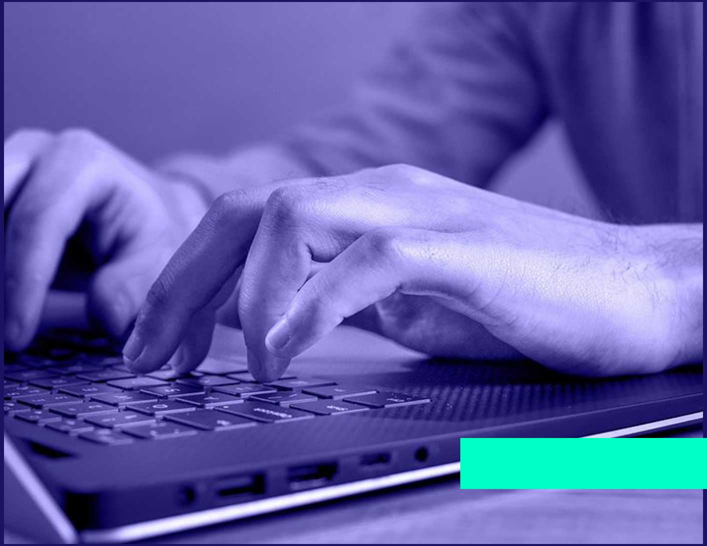
Developer Student Club (MUET)

CONTENTS

1. What is Object Oriented Programming (OOP)?
2. What is a Class?
3. What is an Object?
4. What is an Instance?
5. What are Class Variables?
6. What are Class Methods?
7. QNA
8. Tasks

INTRODUCTION

Object Oriented programming (OOP) is a programming paradigm that relies on the concept of classes and objects. It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects.



“Before software can be reusable it first has to be usable.”

–Ralph Johnson

01

CLASS

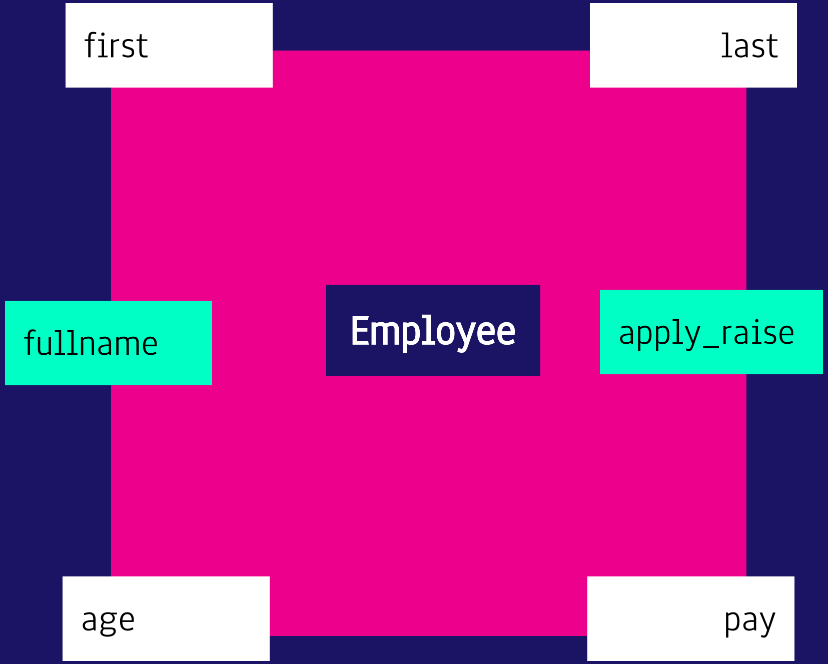
CLASS

A **class** is an abstract blueprint used to create more specific, concrete objects. Classes often represent broad categories, like **Car** or **Employee** that share attributes. These classes define what attributes an instance of this type will have, like **color**, but not the value of those attributes for a specific object.

```
class Employee:
    pass
```

Developer Student Club (MUET)

VISUAL REPRESENTATION



02

OBJECT

OBJECT

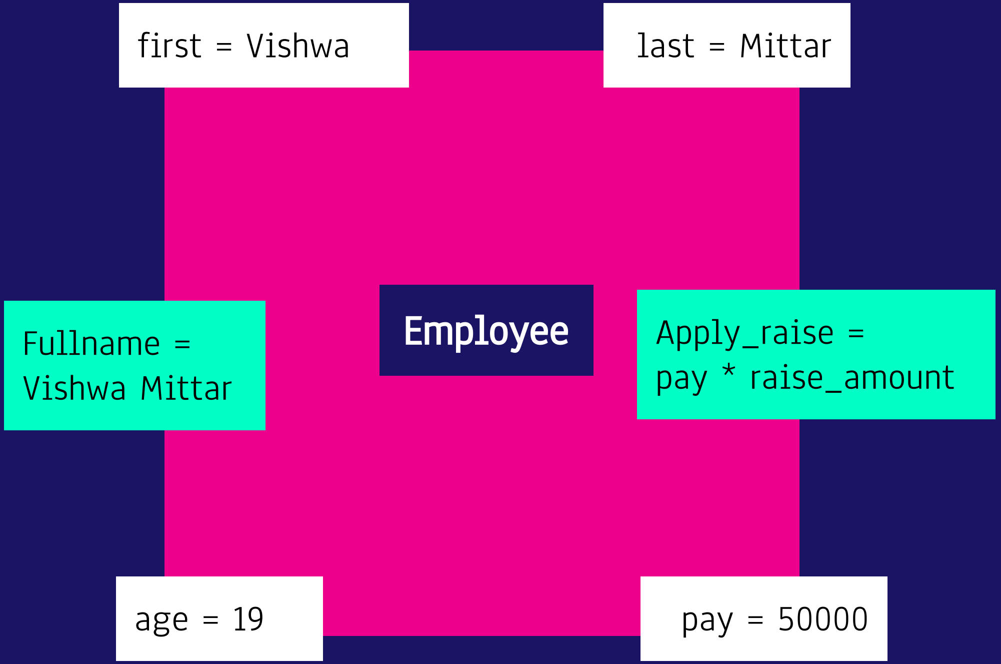
Object is an instance of a class. An object in OOP is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. For example color, shape, name, year, mode, sensors. When you send a message to an object, you are asking the object to invoke or execute one of its methods as defined in the class.

An Object has two characteristics:

Attributes and Behaviour

Employee(“Vishwa”, “Mittar”, 19, 50000)

VISUAL REPRESENTATION



first = Vishwa

last = Mittar

Fullname = Vishwa Mittar

Employee

Apply_raise = pay * raise_amount

age = 19

pay = 50000

03

INSTANCE

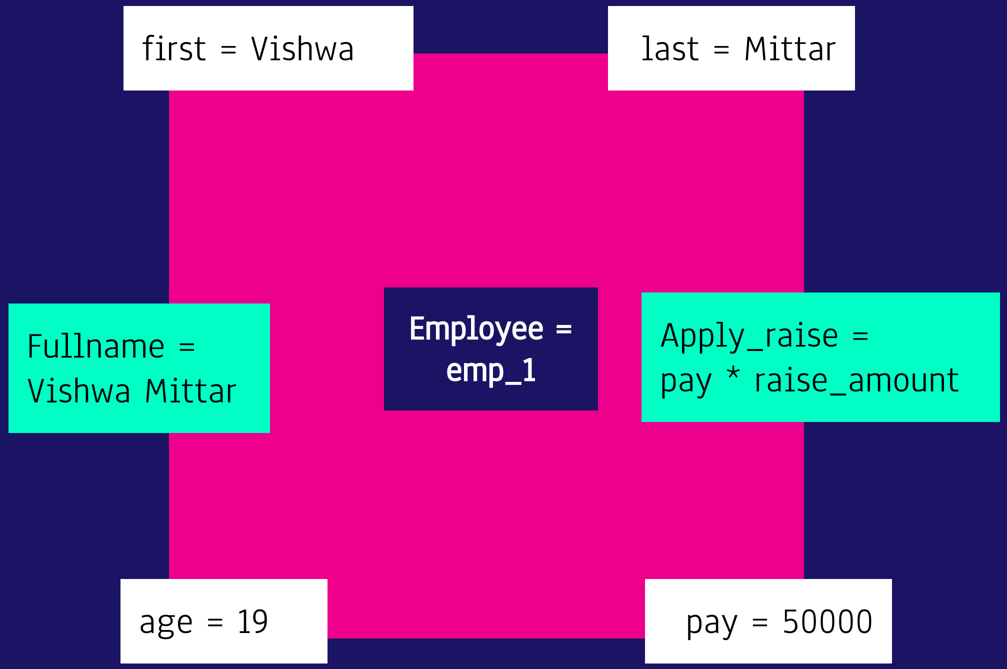
INSTANCE

Object is a generic term , it is physically present but remains undifferentiated. Instance is something that gives them a separate identity.

Simply, reference to object is called an Instance.

```
emp_1 =Employee(“Vishwa”,
“Mittar”, 19, 50000 )
```

VISUAL REPRESENTATION



first = Vishwa

last = Mittar

Fullname = Vishwa Mittar

Employee = emp_1

Apply_raise = pay * raise_amount

age = 19

pay = 50000

04

CLASS VARIABLES

CLASS VARIABLES

A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.
All variables which are assigned a value in the class declaration are class variables. And variables that are assigned values inside methods are instance variables.

```
class Employee:
    raise_amount = 1.04
    no_of_emps = 0
```

05

CLASS METHODS

CLASS METHODS

A special kind of function that is defined in a class definition.

```
class Employee:
        def fullname(self):
                return f’{self.first} {self.last}’
        def apply_raise(self):
                self.pay = int(self.pay * self.raise_amount)
```

FEATURES OF THE OOP

Encapsulation

We can restrict access to methods and variables. This prevents data from direct modification

Polymorphism

Polymorphism is an ability (in OOP) to use a common interface for multiple forms (data types).

Inheritance

Inheritance is a way of creating a new class for using details of an existing class without modifying it.

QNA

Time to hear you

EXERCISE

1. Create a Mobile class without any variables and methods.
2. Create a Mobile class with brand, ram and rom instance attributes.

```
Output: Iphone 4 128
```

3. Using solution of (2) add a method display_specs to the class Mobile which displays specs of mobile.

```
Output: Iphone 7plus, 4GB RAM, 128GB ROM
```

THANKS!

For queries
vishwasluhana@gmail.com
+92-3313148475

vishwas_luhana

Developer Student Club (MUET)