

05feb_assignment

February 14, 2023

Q1. Explain Class and Object with respect to Object-Oriented Programming. Give a suitable example.

In Object-Oriented Programming (OOP), a class is a blueprint, for creating objects, which are instances of the class. A class defines the attributes (data) and methods (behavior) that objects of that class can have.

```
[8]: class car_details:
      def __init__(self,car_name, color, brand):
          self.car_name= car_name
          self.color = color
          self.brand = brand
      def details(self):
          return self.car_name, self.color, self.brand
```

```
[9]: company1 = car_details("suv", "black", "Mahindra")
```

```
[10]: company1.details()
```

```
[10]: ('suv', 'black', 'Mahindra')
```

```
[11]: company2 = car_details("thar","red","mahindra")
```

```
[12]: company2.details()
```

```
[12]: ('thar', 'red', 'mahindra')
```

2. Name the four pillars of OOPs.

Encapsulation, Inheritance, Polymorphism, Abstraction

3. Explain why the **init()** function is used. Give a suitable example.

In Object-Oriented Programming (OOP), the **init()** function is used to initialize the attributes (data) of an object when it is created. This function is also called the constructor

```
[14]: class patient_detail:
      def __init__(self,name,age,phone_number):
          self.name = name
          self.age = age
```

```
self.phone_number = phone_number

def details(self):
    return self.name, self.age, self.phone_number
```

```
[16]: hospital1 = patient_detail("Shubra", 55, 7852369874)
```

```
[18]: hospital1.details()
```

```
[18]: ('Shubra', 55, 7852369874)
```

Q4. Why self is used in OOPs?

In Object-Oriented Programming (OOP), self is a reference to the instance of a class. It is used to access the attributes and methods of the class from within the class itself or from outside the class.

When you create an object from a class, self is used to refer to that object. Each object that is created has its own set of attributes and methods, which are accessed using the self reference.

Q5. What is inheritance? Give an example for each type of inheritance.

Inheritance is a fundamental concept in object-oriented programming (OOP) that allows a new class to be based on an existing class, inheriting its properties (fields and methods) and adding new properties and methods to it. Inheritance promotes code reuse and helps in building complex class hierarchies.

There are several types of inheritance, including:

Single Inheritance: In single inheritance, a class is derived from a single base class. The derived class inherits all the properties and methods of the base class and can add new properties and methods. Example: A class named Square can be derived from a class named Rectangle. The Square class inherits the width and height properties of the Rectangle class, but it adds a new property named side that represents the length of a side.

Multiple Inheritance: In multiple inheritance, a class is derived from multiple base classes. The derived class inherits all the properties and methods of all the base classes. Example: A class named Biped can be derived from two base classes named Human and Animal. The Biped class inherits the properties and methods of both the Human and Animal classes.

Hierarchical Inheritance: In hierarchical inheritance, multiple classes are derived from a single base class. Each derived class inherits all the properties and methods of the base class and can add new properties and methods. Example: A class named Shape can be derived from a base class named Object. Two new classes named Circle and Rectangle can be derived from the Shape class. The Circle and Rectangle classes inherit the properties and methods of the Shape class.

Multilevel Inheritance: In multilevel inheritance, a derived class is derived from another derived class. The new derived class inherits all the properties and methods of the base class and the intermediate derived class. Example: A class named Animal can be derived from a base class named LivingBeing. A new class named Mammal can be derived from the Animal class. A new class named Dog can be derived from the Mammal class. The Dog class inherits the properties and methods of both the Mammal and Animal classes.

[]: