Introduction to OOP

Arif Istiake Sunny

Senior AI Engineer

Brain Station 23 PLC

Mobile: 01732009493

Email: sunny1509006@gmail.com

Contents

Object Oriented Programing (OOP)

Classes

Objects

Introducing Method

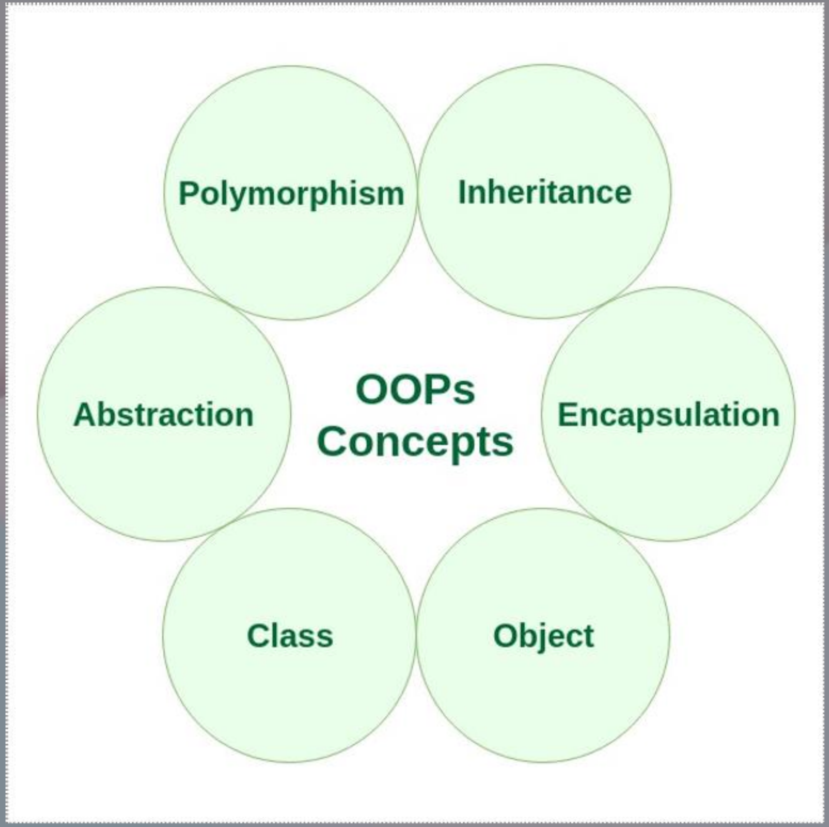
Default Constructors Parameterized Constructor

Pass Statement

Intro to Inheritance

Object Oriented Programming (OOP)

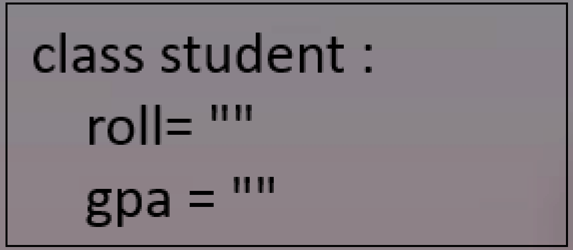
Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.



Classes

Classes are user-defined data types that act as the blueprint for individual objects, attributes and methods

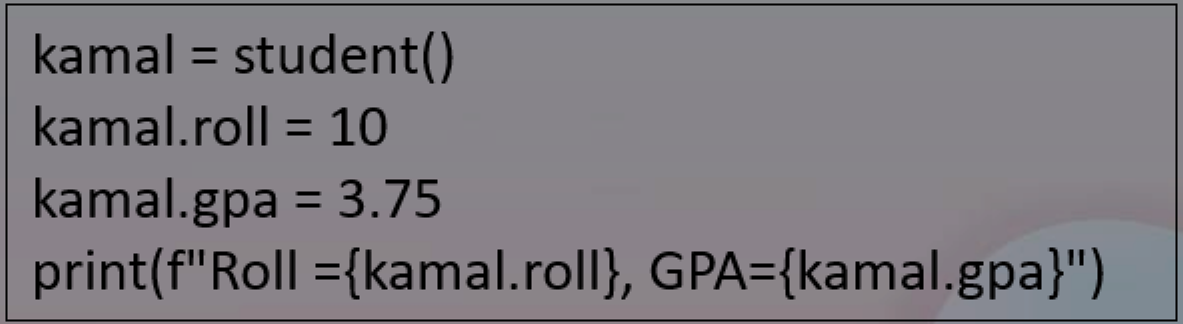
An example of a class is the class Student. Students usually have a roll and gpa; these are attributes.



```
class student :
    roll= ""
    gpa = ""
```

Objects

An Object is an instance of a Class. A class is like a blueprint while an instance is a copy of the class with actual values.



```
kamal = student()
kamal.roll = 10
kamal.gpa = 3.75
print(f"Roll ={kamal.roll}, GPA={kamal.gpa}")
```

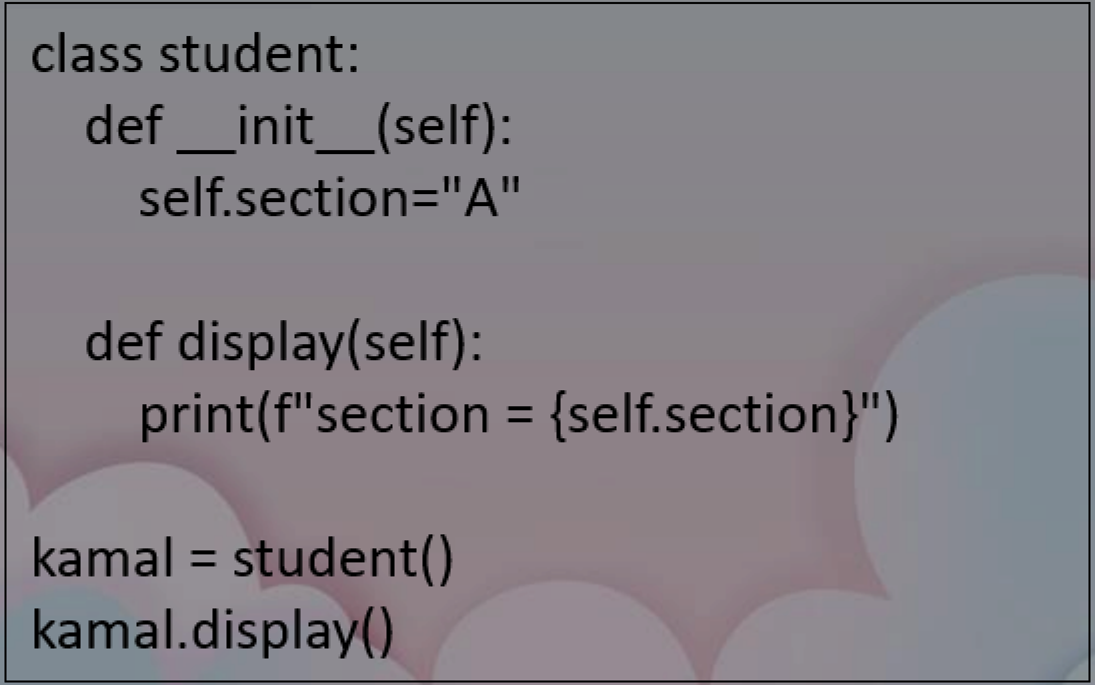
Introducing Method

A method is a function that “belongs to” an object.

```
class student:
    def set_value(self, a, b):
        self.roll = a
        self.gpa = b

    def display(self):
        print(f"Roll ={self.roll}, GPA={self.gpa}")

kamal = student()


kamal.set_value(10, 3.75)
kamal.display()
```

Default Constructors

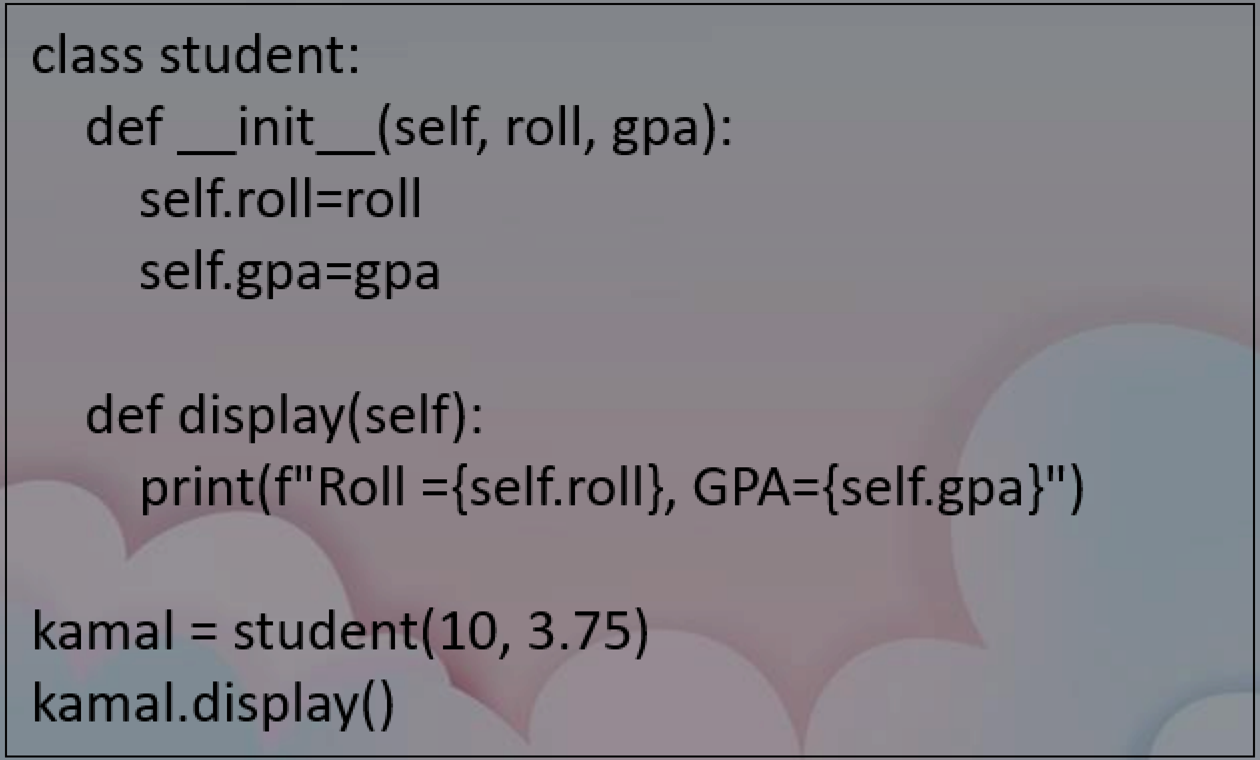
Constructors are generally used for instantiating an object



```
class student:
    def __init__(self):
        self.section="A"

    def display(self):
        print(f"section = {self.section}")

kamal = student()
kamal.display()
```

Parameterized Constructors

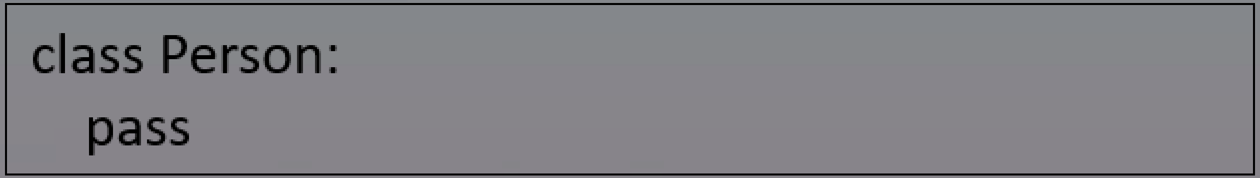
Constructors are generally used for instantiating an object



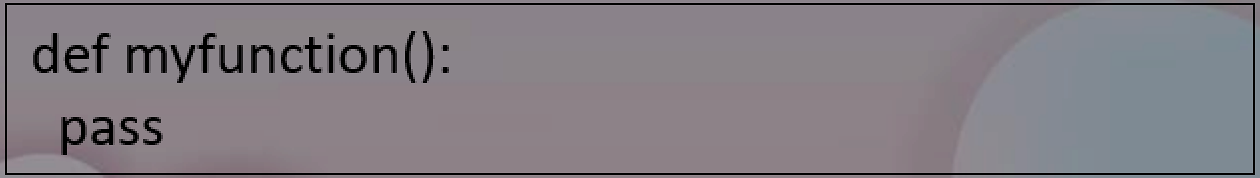
```
class student:
    def __init__(self, roll, gpa):
        self.roll=roll
        self.gpa=gpa

    def display(self):
        print(f"Roll ={self.roll}, GPA={self.gpa}")

kamal = student(10, 3.75)
kamal.display()
```

Pass Statement

Create a placeholder for future code:



```
class Person:
    pass
```



```
def myfunction():
  pass
```

Intro to Inheritance

Inheritance allows us to define a class that inherits all the methods and properties from another class.

Parent class is the class being inherited from, also called base class.

Child class is the class that inherits from another class, also called derived class.

Parent class

```
class Person:
  def __init__(self, fname, lname):
    self.firstname = fname
    self.lastname = lname

  def printname(self):
    print(self.firstname, self.lastname)
```

Child class

```
class Student(Person):
  pass
#-----------------------------------
y = Student("Abul", "Hossain")
y.printname()
```

Avengers: Age of OOP

After the last battle, Tony Stark realized fighting villains alone wasn’t scalable. So, he did what every genius billionaire would do – started a coding bootcamp for the Avengers called "Pyvengers"

What is Object-Oriented Programming (OOP)?

Tony:
"Avengers, listen up! Object-Oriented Programming is like building your own superheroes. You define a blueprint — that's called a **class**. Then you can make multiple versions of that superhero — those are **objects**."

Thor (confused):
"Is Mjölnir an object or a class?"

Tony:
"It's an object, obviously! But if we made many hammers with lightning powers – that would be a class."

Creating a Class & Object

Tony:

"Let’s create a class named `Avenger` and then create objects like `IronMan`, `Hulk`, etc."

Introducing Methods

Tony:
 "A **method** is like a superhero's ability. You attach it to the class, and every object can use it."

Thor:
 "And I shall introduce myself with thunder!"

Default Constructor

Tony:

"When you want to set something by default when an object is created, you use a constructor."

Steve:

"Avengers... Initialized!"

Parameterized Constructor

Tony:
"But if you want to customize your Avenger while creating them…"

Spider-Man:
"Finally! I’m not just the intern anymore."

Pass Statement

Tony:

"And sometimes, you just want a placeholder – like Hawkeye in the final battle. You use `pass`."

Hawkeye:

"Hey! I was useful... kinda."

Intro to Inheritance

Tony:
"Now comes the big stuff — **inheritance**. Suppose all Avengers are heroes. Let’s build a base class `Hero`, and let other heroes inherit its powers."

Thor:
"So my powers can be... inherited?"

Tony:
"Well, not the god complex — but yes."

🏁 Homework (For Students)

- Create a `Villain` class and inherit from it to create `Loki` and `Ultron`.
- Add different methods for each villain.
- Use constructors to initialize their evil plans.

🔚 Closing Line (From Tony)

"Remember team – in OOP, with great **classes** comes great **objects**!"

Thank You