

# oop1

December 25, 2022

## 1 Object Oriented Programmng - 1

## 2 Introduction to OOP | Class & Object

- OPP
- Class & Method
- Verification whether object created under class
- Class access by object
- Print
- Method
- Constructor

### Assumptions

- First Create a class by “class” keyword
- Class name is ‘Student’ first Capital Letter
- Under the class define some common properties / variables/ features / class members e.g roll & gpa

## 3 Define Class

```
[22]: class Student:
      name = " "
      roll = " "
      gpa = " "
```

- Now Need to create class objects before using this

## 4 Object Creation

```
[23]: rahim = Student()    # Creating an object (rahim) under student class
      karim = Student()    # Creating 2nd object
      reza = Student()     # Creating 3rd object
```

### 4.0.1 Verification whether rahim created as object under Student Class

- isinstance is a method / function that will return a boolean number to show wether object created under a class

- Once object created successfully, then the object (rahim) will get all the class access

```
[5]: print(isinstance(rahim, Student))
```

True

## 4.1 Object Using Class Access

- An object can access the class by using dot operator

```
[24]: rahim.name = 'Rahim'
      rahim.roll = 101
      rahim.gpa = 3.50
```

```
[25]: karim.name = 'Karim'
      karim.roll = 102
      karim.gpa = 3.00
```

```
[26]: reza.name = 'Reza'
      reza.roll = 103
      reza.gpa = 3.25
```

## 5 Print

```
[27]: print (f"Name : {rahim.name}, Roll: {rahim.roll}, GPA :{rahim.gpa} "),
      print (f"Name : {karim.name}, Roll: {karim.roll}, GPA :{karim.gpa} "),
      print (f"Name : {reza.name}, Roll: {reza.roll}, GPA : {reza.gpa} ")
```

Name : Rahim, Roll: 101, GPA :3.5  
 Name : Karim, Roll: 102, GPA :3.0  
 Name : Reza, Roll: 103, GPA : 3.25

## 6 Introducing Method

### 6.0.1 Method Creation / Function under Class

- Self is a param inside display function/ method

```
[39]: class Student1:
      roll = " "
      gpa = " "

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

## 6.0.2 Object Creation

```
[50]: samim = Student1()    # Creating an object (rahim) under student class
      suman = Student1()    # Creating 2nd object
      sohel = Student1()    # Creeating object of Sohel
```

## 6.0.3 Value Assign to object / Using object

```
[43]: samim.roll = 17
      samim.gpa = 3.15
```

```
[44]: suman.roll = 18
      suman.gpa = 3.18
```

```
[45]: sohel.roll = 19
      sohel.gpa = 2.90
```

## 6.0.4 Print / Display by using method or created function

- Calling “Display” function by “Object” with dot operator

```
[47]: samim.display()
```

Roll : 17, GPA : 3.15

```
[48]: suman.display()
```

Roll : 18, GPA : 3.18

```
[49]: sohel.display()
```

Roll : 19, GPA : 2.9

## 7 Exercise 3: Automatic Value Assign for all, not explicitly value assign

- Creating another function & call them & that function will set Roll & GPA
- Give a function name e.g set\_value (any name can be provided), where two param will be passed as ‘Roll’ & ‘GPA’

```
[51]: class Student3:
      roll = " "
      gpa = " "

      def set_value (self, roll, gpa):
          self.roll = roll
          self.gpa = gpa

      def display (self):
```

```
print (f"Roll : {self.roll}, GPA : {self.gpa}")
```

### 7.0.1 Object Creation

- Object Creation under Class

```
[58]: tansim = Student3()  
      tahia = Student3 ()  
      farin = Student3()
```

#### Set Value & Print

```
[56]: tansim.set_value(1, 5.00)  
      tansim.display()
```

Roll : 1, GPA : 5.0

```
[57]: tahia.set_value(2, 5.00)  
      tahia.display()
```

Roll : 2, GPA : 5.0

```
[59]: farin.set_value(5, 4.50)  
      farin.display()
```

Roll : 5, GPA : 4.5

## 8 Constructor

- Value passing during object creation
- As a result, no need to created fucntion to set \_values
- There is method called as “**init**” required to create constructor
- Make sure ‘self’ must in every method
- Constructor is a SPECIAL type of METHOD, that doesn’t requirs calling explicitly
- Contruction helps to initialize object (or help to set value during object creation)

#### Class Cretion

```
[60]: class Student4 :  
      roll = " "  
      gpa = " "  
  
      def __init__(self, roll, gpa):  
          self.roll = roll  
          self.gpa = gpa  
  
      def display(self):  
          print(f"Roll : {self.roll}, GPA : {self.gpa} ")
```

#### Object Creation with value initialize

```
[63]: riyaad = Student4 (25, 5.00) # can passing values during object declaration
      ↪ due to constructor
      riyaad.display()
```

Roll : 25, GPA : 5.0

```
[64]: rifat = Student4 (26, 5.00)
      rifat.display()
```

Roll : 26, GPA : 5.0

```
[65]: john = Student4 (27, 3.50)
      john.display()
```

Roll : 27, GPA : 3.5