# Assignment- 9

## ELP - 718 Telecom Software Laboratory

**Ch Krishna Chaitanya**
**2019JTM2674**
**2019-21**

A report on Python with OOPs concepts


**Bharti School of**
**Telecommunication Technology and Management**
**IIT Delhi**
**India**

October 9, 2019

# Contents

# Objective Statement

To test our understanding of object oriented concepts using python.

# 1 Problem Statement -1

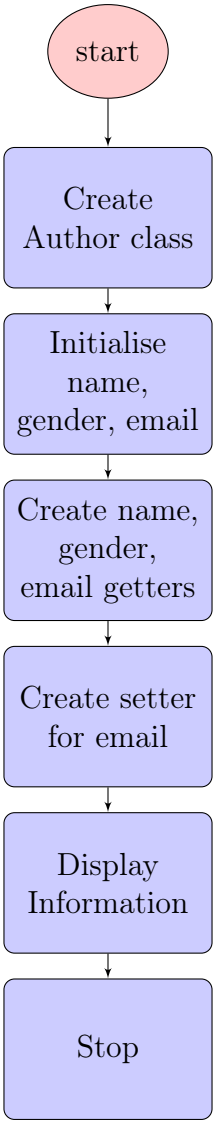
To ceate class named Author which holds data members: name (string), email (string), and gender

## 1.1 Problem Satement

## 1.2 Algorithm and Implementation

- Create Class Author
- Initialise constructor with attribures name, gender, email
- Name and Gender cannot be changed hence only getters
- Email can be updated or changed using setter function
- Implement print method in Author class
- Display details using print method in Author class

## 1.3 Flowchart




## 1.4 Input and Output format

### 1.4.1 Input Format

Create object with Author Class

### 1.4.2 Output Format

Peter Jones (m) at peter@somewhere.com

## 1.5 Test Cases

### 1.5.1 Input1

person1=Author("Krishna","kc@gmail.com","m")

### 1.5.2 Output1

Krishna (m) at kc@gmail.com

## 1.6 Screenshots

# 2 Problem Statement -2

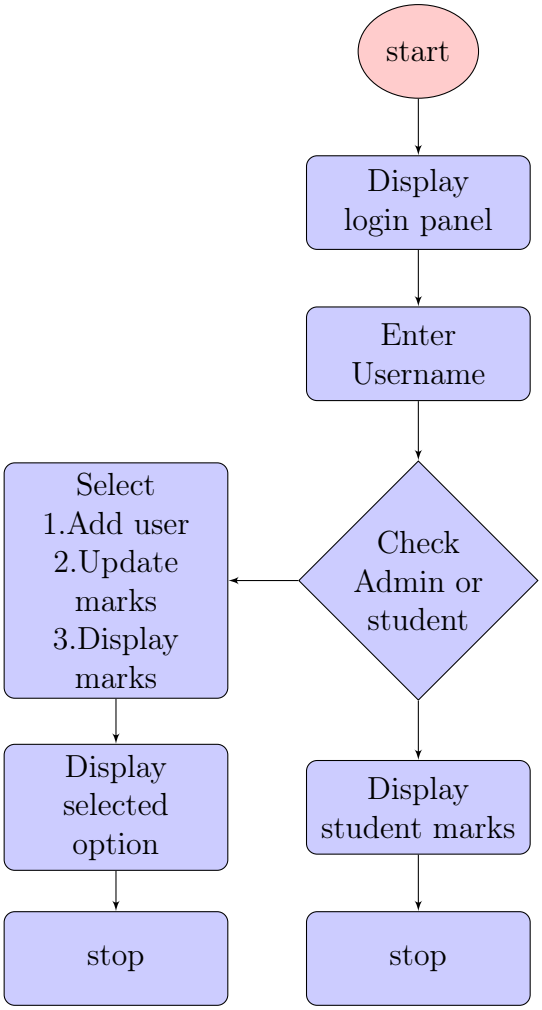
## 2.1 Problem Satement

To make an application to handle the students results using classes in python

## 2.2 Algorithm and Implementation

- Login panel for student and admin should be created
- Validate password for admin
- If user is student, display his marks
- If user is admin, three options are available
- Display required data based on option chosen
- Check if he wants to exit or continue

## 2.3 Flowchart



## 2.4 Test Cases

### 2.4.1 Input

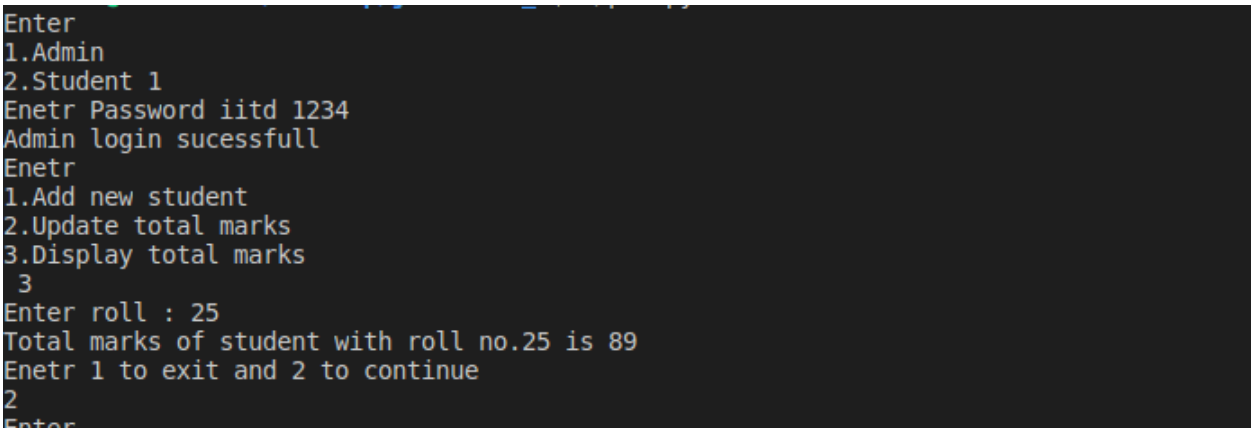
Enter
1.Admin
2.Student 2

### 2.4.2 Output

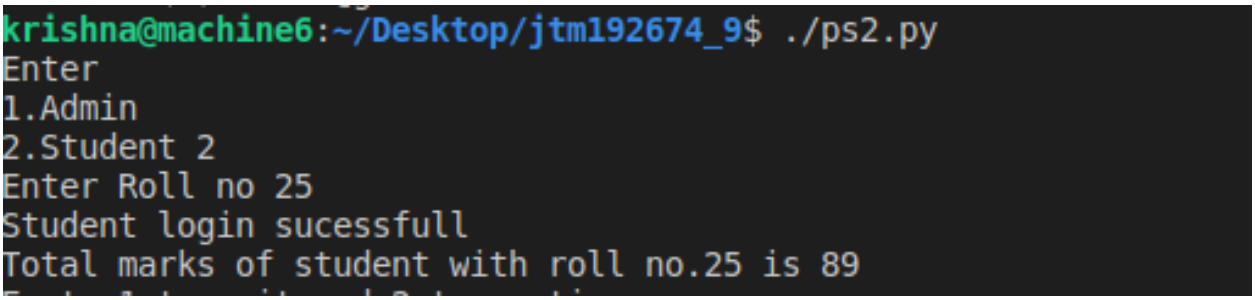
Enter
1.Admin
2.Student 2
Enter Roll no 25
Student login succesfull
Total marks of student with roll no.25 is 89

## 2.5 Screenshots

### 2.5.1 Screenshot1



### 2.5.2 Screenshot2



# Appendices

## Problem 1

### code:

```
#!/usr/bin/python3

#Creation of class Author
class Author:
    def __init__(self,name,email,gender):   #initialising function
        self.name=name
        self.email=email
        self.gender=gender

    def get_name(self):                     #Getter for name
        return self.name

    def get_geder(self):                    #Getter for gender
        return self.gender

    def set_email(self,new_email):          #Setter for email
        self.email=new_email

    def get_email(self):                    #Getter for email
        return self.email

    def printinfo(self):                    #print function
        print("{} ({}) at {}".format(self.name,self.gender,self.email))




#creating instance of Author object
person1=Author("Krishna","kc@gmail.com","m")
person1.printinfo()
```

## Problem 2

### code:

```
#!/usr/bin/python3
#Program to make an application to handle student's results
#Creation of class Student
class Student:
    def __init__(self,name,roll):
        self.name=name
        self.roll=roll

#Creation of class Student_Base
class Student_Base:
    totalMarks=89
    def __init__(self,roll):
        self.roll=roll

    #Add new student
    def addNewStudent(self,new_name):
        self.new_name=new_name
        #creating instance of Student object
        n1=Student(self.new_name,self.roll)
        print(n1.__dict__)
        print("Student added successfully")

    #Update total marks
    def displayMarks(self):
        print("Total marks of student with roll no.{} is {}".format(self.roll,Student_Ba

    #Display total marks\n
    def updateMarks(self,update_marks):
        print("Total marks of student with roll no.{} is {}".format(self.roll,Student_Ba
        #updating marks
        Student_Base.totalMarks=update_marks
        print("Updated marks of student with roll no.{} is {}".format(self.roll,Student_

while 1:
    #creating instance of Student object
    student1=Student("krishna","25")
    flag = 0
    user=int(input("Enter\n1.Admin\n2.Student "))

    #User Validation
    if user == 1:                                                    #Admin validation
        password=input("Enetr Password ")
```

```
    flag=1
    if password == "iitd_1234":
        print("Admin login succesfull")
        #Student Validation
elif user == 2:
    stu_roll = input("Enter Roll no ")
    if stu_roll == "25":
        print("Student login succesfull")
        s1=Student_Base(stu_roll)
        #Display Student Marks
        s1.displayMarks()

#Functionality for admin
if flag == 1:
    option=int(input("Enetr\n1.Add new student\n2.Update total marks\n3.Display tota

    #Add new student
    if option == 1:
        new_name=input("Enter Student Name: ")
        new_roll=input("Enter Roll: ")
        a3=Student_Base(new_roll)
        a3.addNewStudent(new_name)
        new_student=Student(new_name,new_roll)

    #Update total marks
    elif option == 2:
        update_roll=input("Enetr roll number : ")
        a1=Student_Base(update_roll)
        a1.totalMarks = int(input("Enter updated marks : "))
        a1.updateMarks(a1.totalMarks)

    #Display total marks\n
    elif option == 3:
        disp_roll = input("Enter roll : ")
        a2=Student_Base(disp_roll)
        a2.displayMarks()



#Asking to exit or continue
x=int(input("Enetr 1 to exit and 2 to continue\n"))
if x==1:
```

```
    break
elif x==2:
    continue
```

# References


[1] Flowchart using Latex
Kjell Magne Fauske
http://www.texample.net/tikz/examples/simple-flow-chart/

[2] Python Basics
https://docs.python.org/3/

[3] Python OOPS Concepts
https://code.tutsplus.com/articles/python-from-scratch-object-oriented-programming--net-21476

[4] Git Hub
https://help.github.com/en/articles/fork-a-repo

[5] Python Getters and Setters
https://www.datacamp.com/community/tutorials/property-getters-setters