

~\Downloads\practice.ipynb - Colab\_files\practicework.py

```
1 #Define a class Student with attributes name, age, score. Create an instance and print its
2 # attributes.
3 class Student:
4     def __init__(self, name, age, score):
5         self.name = name
6         self.age = age
7         self.score = score
8     student1 = ("Hassan", 15, 85)
9     print("Name:", student1.name)
10    print("Age:", student1.age)
11    print("Score:", student1.score)
12 #Add a method display_info() to Student to print a sentence about the student.
13 class Student:
14     def display_info(self):
15         print("we are learning python")
16 #Define a method has_passed() that returns True if score >= 50, else False.
17 def has_passed(score):
18     return score >= 50
19 print(has_passed(75))
20 print(has_passed(40))
21 #Create a second class Teacher with name, subject, and a method teach(). Instantiate and call
22 # its method.
23 class Teacher:
24     def __init__(self, name, subject):
25         self.name = name
26         self.age = age
27     def teach(self):
28         print(f"{self.name} is teaching {self.subject}.")
29 t1 = Teacher("Ms. Mubaraka", Computer)
30 #Modify Student class to include a class variable school = "GUJAR AI School" and print it from
31 # instances.
32 class student:
33     school = "GUJAR AI School"
34 s = student()
35 print("s.school")
36 #Create two student objects and compare if they are instances of Student using isinstance().
37 class Student:
38     pass
39 s1 = Student()
40 s2 = Student()
41 print(isinstance(s1, Student))
42 print(isinstance(s2, Student))
43 #Create a Course class with attributes title and students (a list). Add a method to add
44 # students.
45 class Course:
46     def __init__(self, title):
47         self.title = title
48         self.students = []
```

```
45     def add_student(self, student):
46         self.students.append(student)
47 #Write a method in Course to compute the average score of enrolled students.
48 class Course:
49     def __init__(self, title):
50         self.title = title
51         self.students = []
52     def add_student(self, score):
53         self.students.append(score)
54     def average_score(self):
55         return sum(self.students) / len(self.students)
56 #Override the __str__() method in Student class to return a custom string.
57 class Student:
58     def __init__(self, name):
59         self.name = name
60     def __str__(self):
61         return f"Student name is {self.name}"
62 #Use the dir() function to list attributes & methods of an instance of any class.
63 class Student:
64     def __init__(self, name):
65         self.name = name
66 s = Student("Sarmad")
67 print(dir(s))
68
```