

学号	姓名	提交时间	总分	成绩
-	孙雨婷	2023-05-09 19:18:59	100	94

1. 【简答题】设计一个基于Python面向对象思想的"Student"类, 具备以下属性和方法 :

得分 : 94/100

属性 :

- name : 学生姓名

- age : 学生年龄

- gender : 学生性别

- score : 学生分数

方法 :

- init(self, name, age, gender) : 学生初始化方法, 初始化以上四个属性

- set_score(self, score) : 设置学生分数

- get_score(self) : 返回学生分数, 如果学生未设置分数, 则返回"暂未设置分数"提示信息

要求 :

请编写程序, 根据以上要求设计并实现"Student"类, 并创建两个学生对象, 分别设置他们的分数并进行输出。每个输出语句格式如下 :

[学生姓名]的分数是[学生分数]

学生答案 : class Students:

```
def __init__(self, name, age):
```

```
self.name = name
```

```
self.age = age
```

```
self.gander = "男"
```

```
self.score = None
```

```
def set_score(self, score):
```

```
if score < 0:
```

```
print("分数不能小于0")
```

```
elif score > 100:
```

```
print("分数不能大于100")
```

```
self.score = score
```

```
def get_score(self):
```

```
if self.score is None:
```

```
print("暂未设置分数")
```

```
else:
```

```
print(self.score)
```

```
LiLei = Students("李雷", 18)
```

```
LiLei.score = 88
```

```
print("{}的分数是{}".format(LiLei.name,LiLei.score))
```

```
HanMeiMei = Students("韩梅梅",16)
```

```
HanMeiMei.score = 99
```

```
HanMeiMei.gander = "女"
```

```
print("{}的分数是{}".format(HanMeiMei.name,HanMeiMei.score))
```

参考答案 : 略