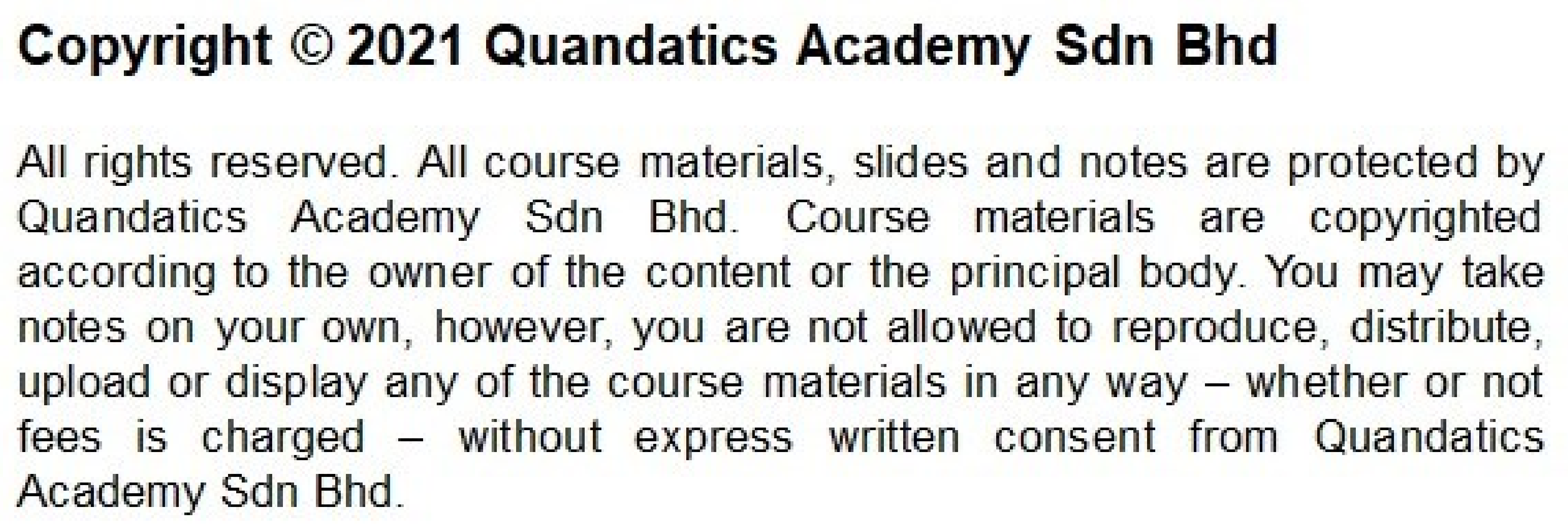


Class

Trainer : Foo Chee Chuan

Class

UTM (Johor)

→

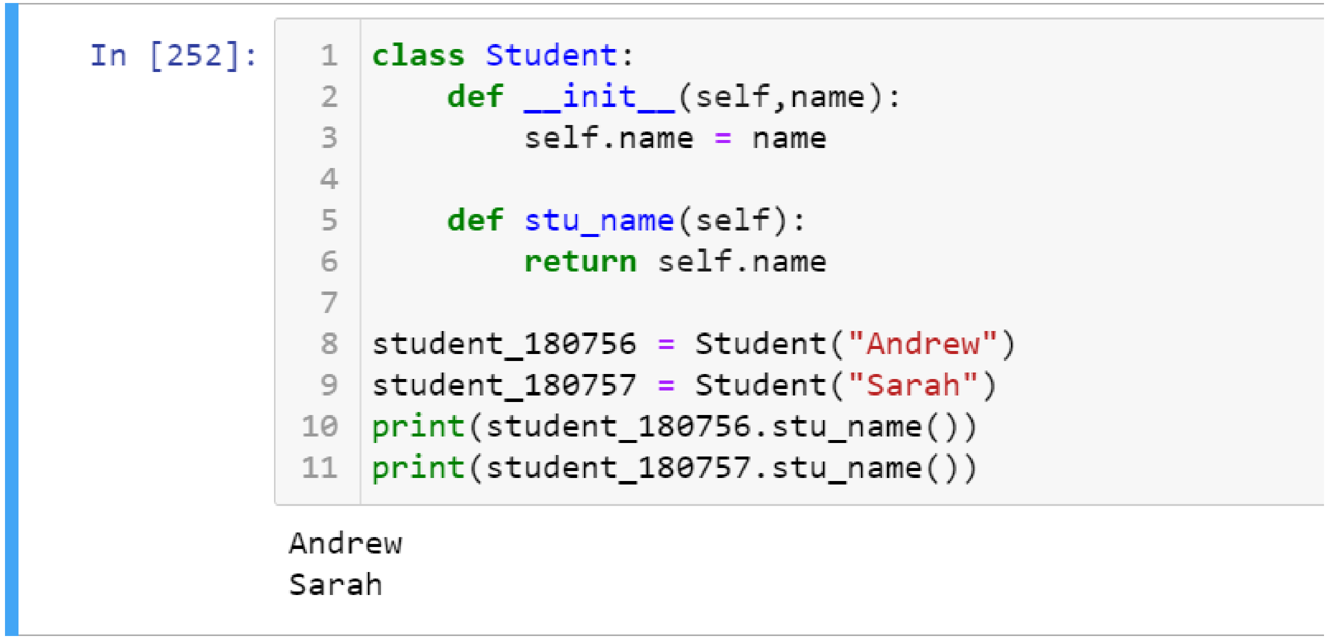
Students

- StudentID
- Name

Initialization

__init__(self)

Identify each students



In [252]:

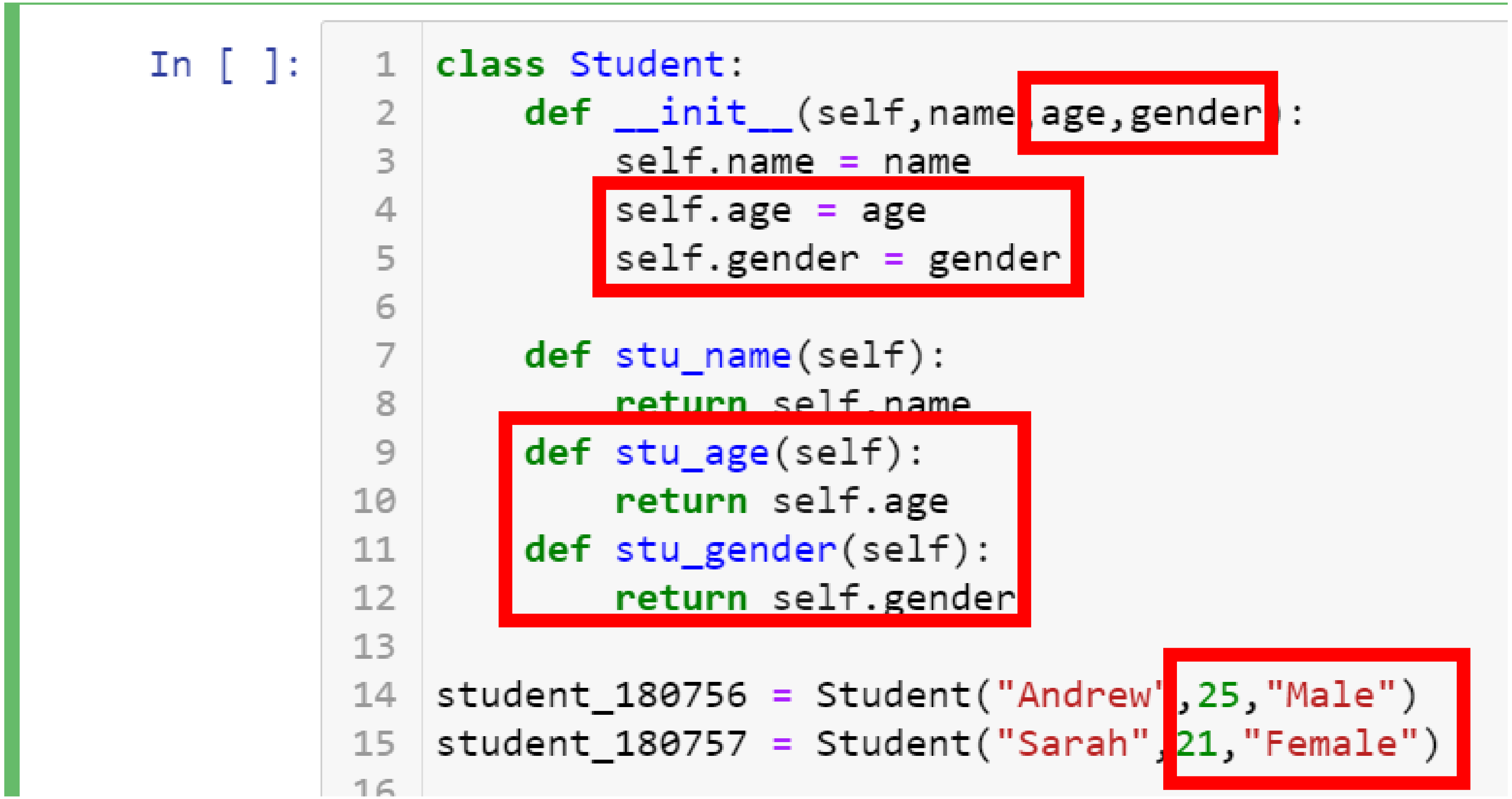
```python
class Student:
    def __init__(self,name):
        self.name = name

    def stu_name(self):
        return self.name

student_180756 = Student("Andrew")
student_180757 = Student("Sarah")
print(student_180756.stu_name())
print(student_180757.stu_name())
```

```
Andrew
Sarah
```

- **Self : StudentID**
- **Self.name : Name of the student belong to this studentID**
- **1 object = 1 student = 1 student ID**

Instance Variables

● StudentID : 180756	● StudentID : 180757
● Name : Andrew	● Name : Sarah
● Age : 25	● Age : 21
● Gender : Male	● Gender : Female



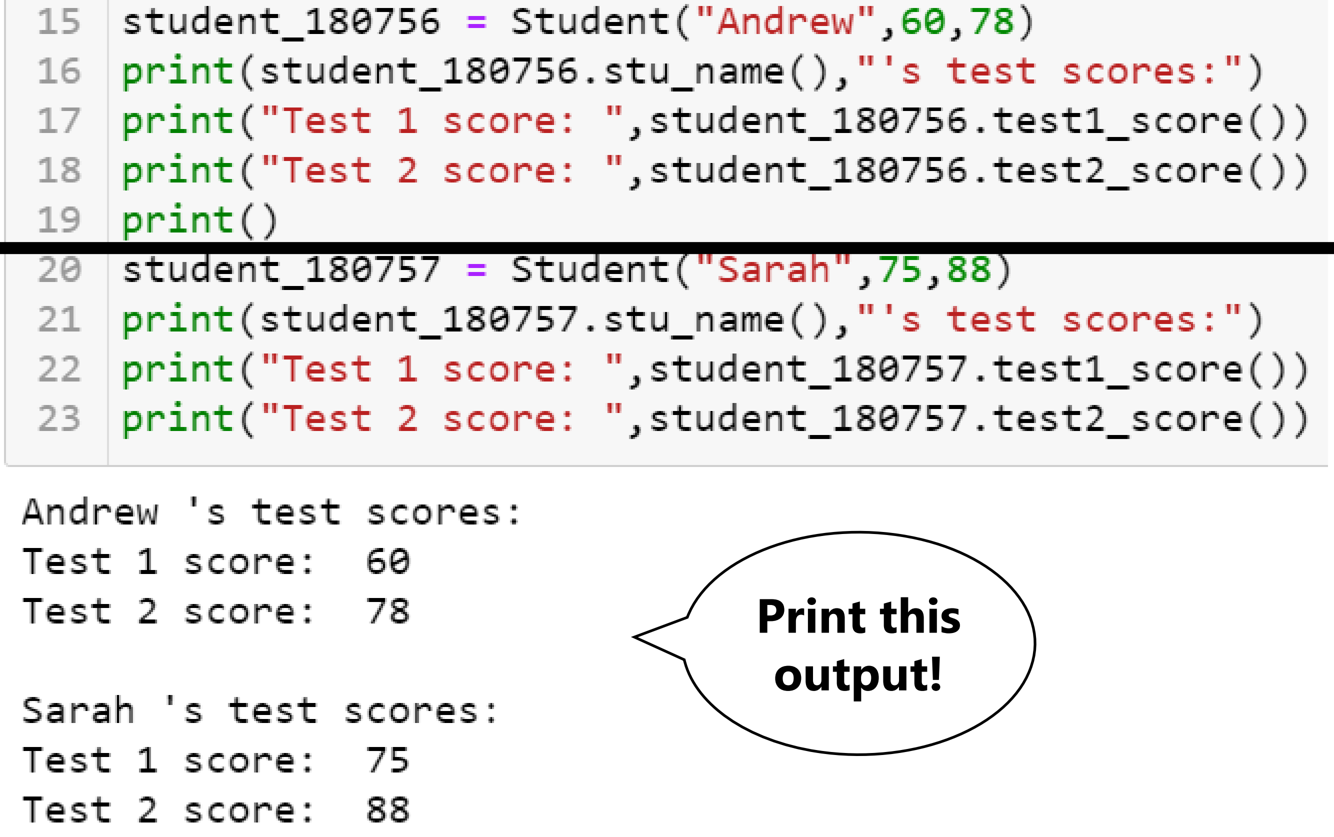
```
In [ ]:
class Student:
    def __init__(self,name,age,gender):
        self.name = name
        self.age = age
        self.gender = gender

    def stu_name(self):
        return self.name
    def stu_age(self):
        return self.age
    def stu_gender(self):
        return self.gender

student_180756 = Student("Andrew",25,"Male")
student_180757 = Student("Sarah",21,"Female")
```

Practice 1: Print Test Scores

- Test_01 : 60%
- Test_02 : 78%

- Test_01 : 75%
- Test_02 : 88%

Create an instance variable for test_01 & test_02 respectively



```python
student_180756 = Student("Andrew",60,78)
print(student_180756.stu_name(),"'s test scores:")
print("Test 1 score: ",student_180756.test1_score())
print("Test 2 score: ",student_180756.test2_score())
print()
student_180757 = Student("Sarah",75,88)
print(student_180757.stu_name(),"'s test scores:")
print("Test 1 score: ",student_180757.test1_score())
print("Test 2 score: ",student_180757.test2_score())
```

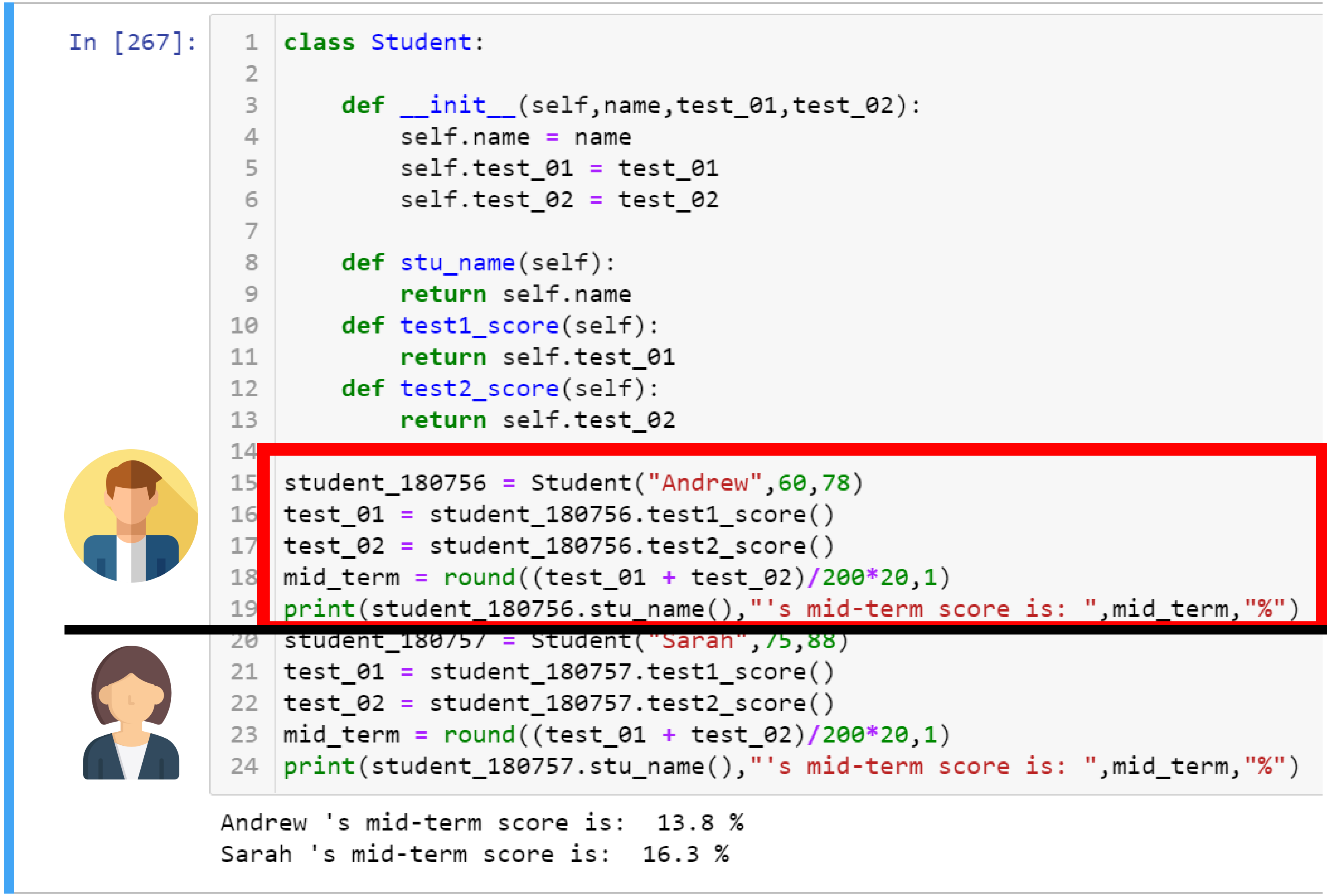
```
Andrew 's test scores:
Test 1 score:  60
Test 2 score:  78

Sarah 's test scores:
Test 1 score:  75
Test 2 score:  88
```

Print this output!

Calculate test score

- **Mid-term : test 1 + test 2 = (20%)**
- **Assignment : (20%)**
- **Final : (60%)**

Mid-term = (test 1 + test 2)/200*20



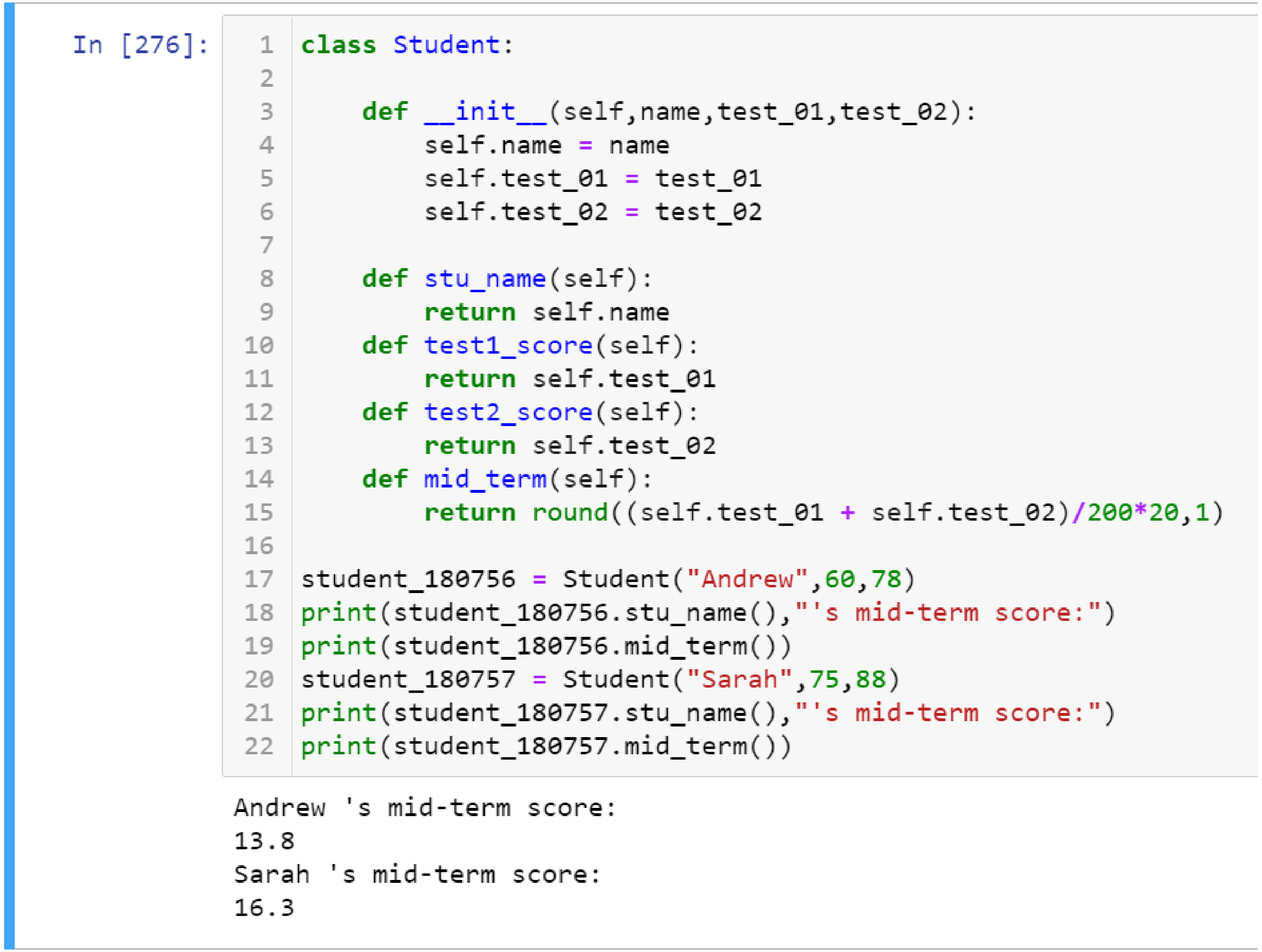
In [267]:

```
class Student:

    def __init__(self,name,test_01,test_02):
        self.name = name
        self.test_01 = test_01
        self.test_02 = test_02

    def stu_name(self):
        return self.name
    def test1_score(self):
        return self.test_01
    def test2_score(self):
        return self.test_02

student_180756 = Student("Andrew",60,78)
test_01 = student_180756.test1_score()
test_02 = student_180756.test2_score()
mid_term = round((test_01 + test_02)/200*20,1)
print(student_180756.stu_name(),"'s mid-term score is: ",mid_term,"%")
student_180757 = Student("Sarah",75,88)
test_01 = student_180757.test1_score()
test_02 = student_180757.test2_score()
mid_term = round((test_01 + test_02)/200*20,1)
print(student_180757.stu_name(),"'s mid-term score is: ",mid_term,"%")
```

```
Andrew 's mid-term score is:  13.8 %
Sarah 's mid-term score is:  16.3 %
```

Instance Method

Midterm score Method



In [276]:

```python
class Student:

    def __init__(self,name,test_01,test_02):
        self.name = name
        self.test_01 = test_01
        self.test_02 = test_02

    def stu_name(self):
        return self.name
    def test1_score(self):
        return self.test_01
    def test2_score(self):
        return self.test_02
    def mid_term(self):
        return round((self.test_01 + self.test_02)/200*20,1)

student_180756 = Student("Andrew",60,78)
print(student_180756.stu_name(),"'s mid-term score:")
print(student_180756.mid_term())
student_180757 = Student("Sarah",75,88)
print(student_180757.stu_name(),"'s mid-term score:")
print(student_180757.mid_term())
```

```
Andrew 's mid-term score:
13.8
Sarah 's mid-term score:
16.3
```

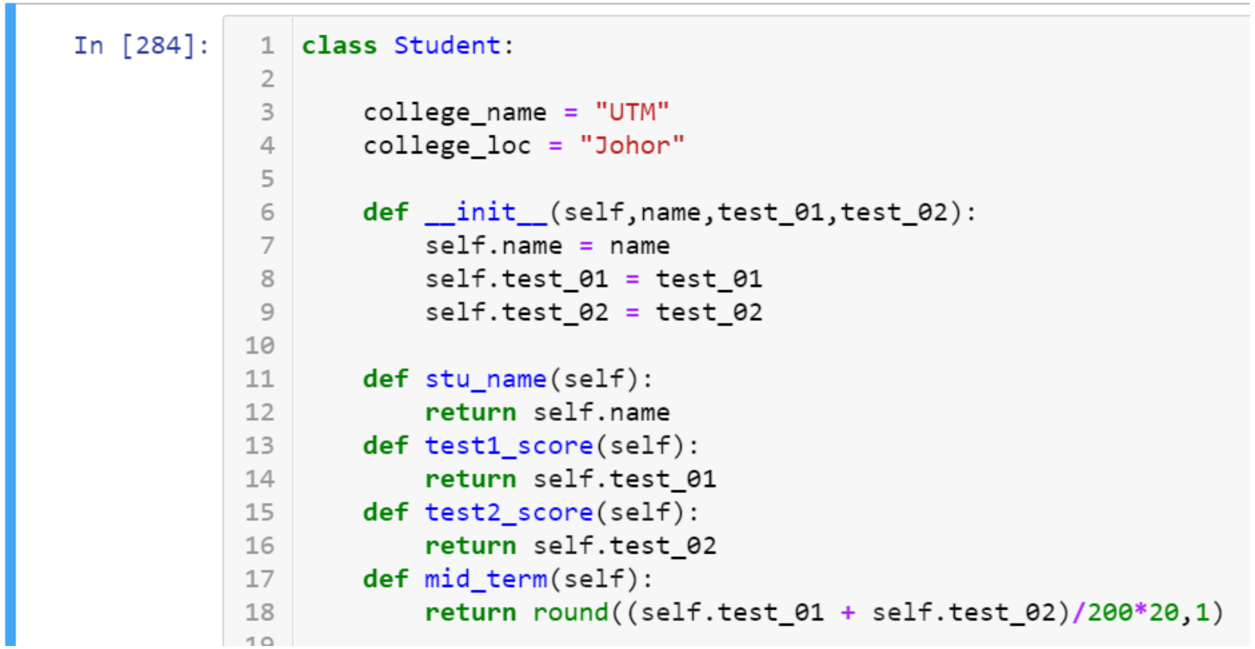
Class Methods

UTM (Johor)

→

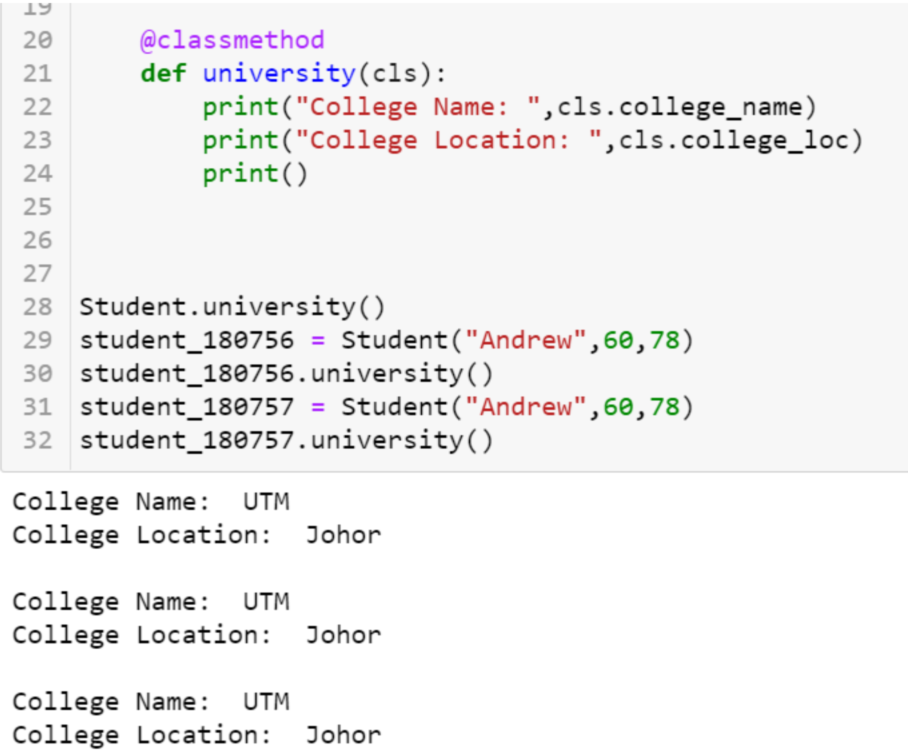
Students

Class Methods



In [284]:

```python
class Student:

    college_name = "UTM"
    college_loc = "Johor"

    def __init__(self,name,test_01,test_02):
        self.name = name
        self.test_01 = test_01
        self.test_02 = test_02

    def stu_name(self):
        return self.name
    def test1_score(self):
        return self.test_01
    def test2_score(self):
        return self.test_02
    def mid_term(self):
        return round((self.test_01 + self.test_02)/200*20,1)
```

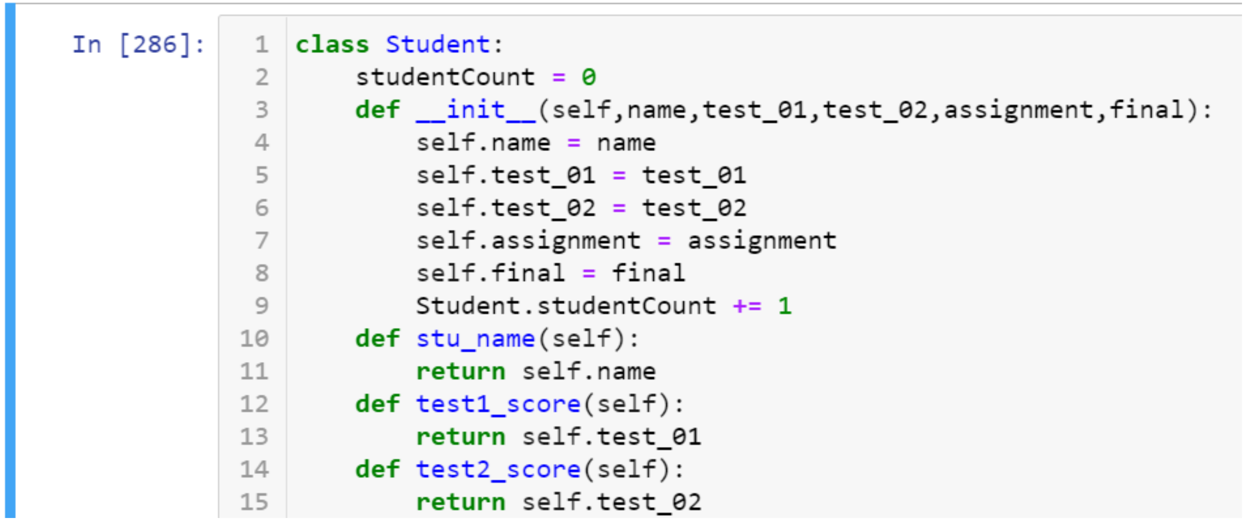


```python

    @classmethod
    def university(cls):
        print("College Name: ",cls.college_name)
        print("College Location: ",cls.college_loc)
        print()



Student.university()
student_180756 = Student("Andrew",60,78)
student_180756.university()
student_180757 = Student("Andrew",60,78)
student_180757.university()
```

```
College Name:  UTM
College Location:  Johor

College Name:  UTM
College Location:  Johor

College Name:  UTM
College Location:  Johor
```

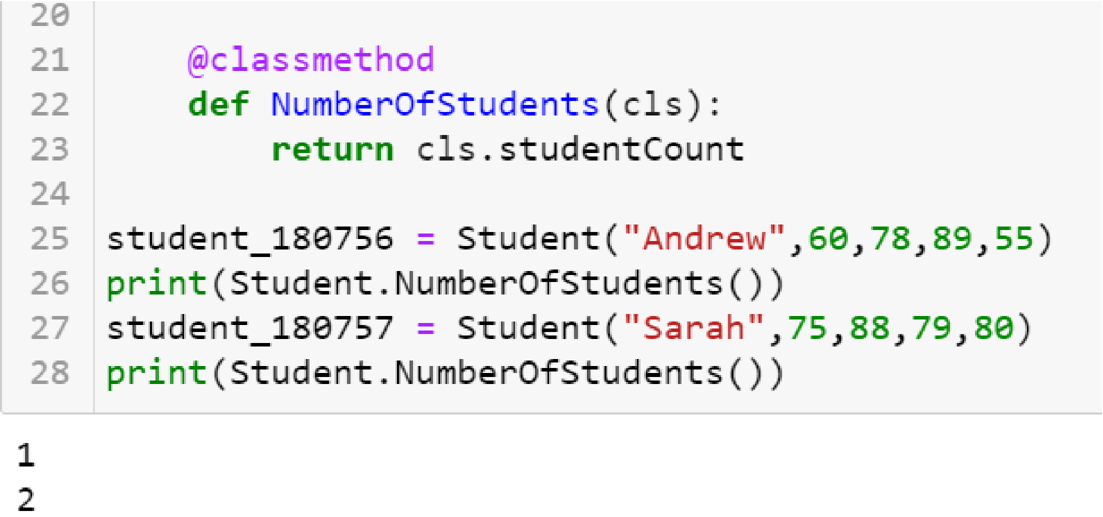
Another Example

Student Count (Class Method)



In [286]:

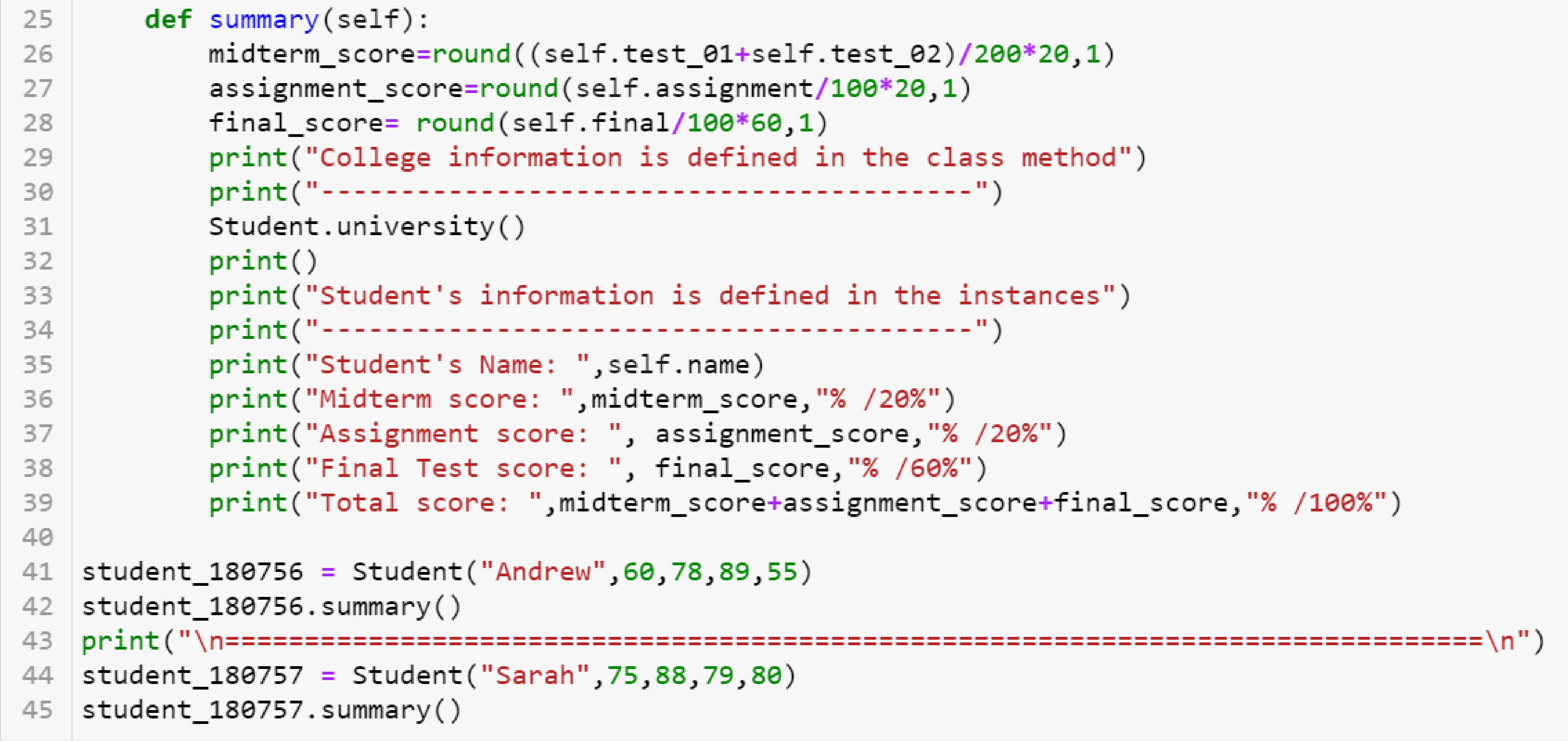
```python
class Student:
    studentCount = 0
    def __init__(self,name,test_01,test_02,assignment,final):
        self.name = name
        self.test_01 = test_01
        self.test_02 = test_02
        self.assignment = assignment
        self.final = final
        Student.studentCount += 1
    def stu_name(self):
        return self.name
    def test1_score(self):
        return self.test_01
    def test2_score(self):
        return self.test_02
```



```python

    @classmethod
    def NumberOfStudents(cls):
        return cls.studentCount

student_180756 = Student("Andrew",60,78,89,55)
print(Student.NumberOfStudents())
student_180757 = Student("Sarah",75,88,79,80)
print(Student.NumberOfStudents())
```

```
1
2
```

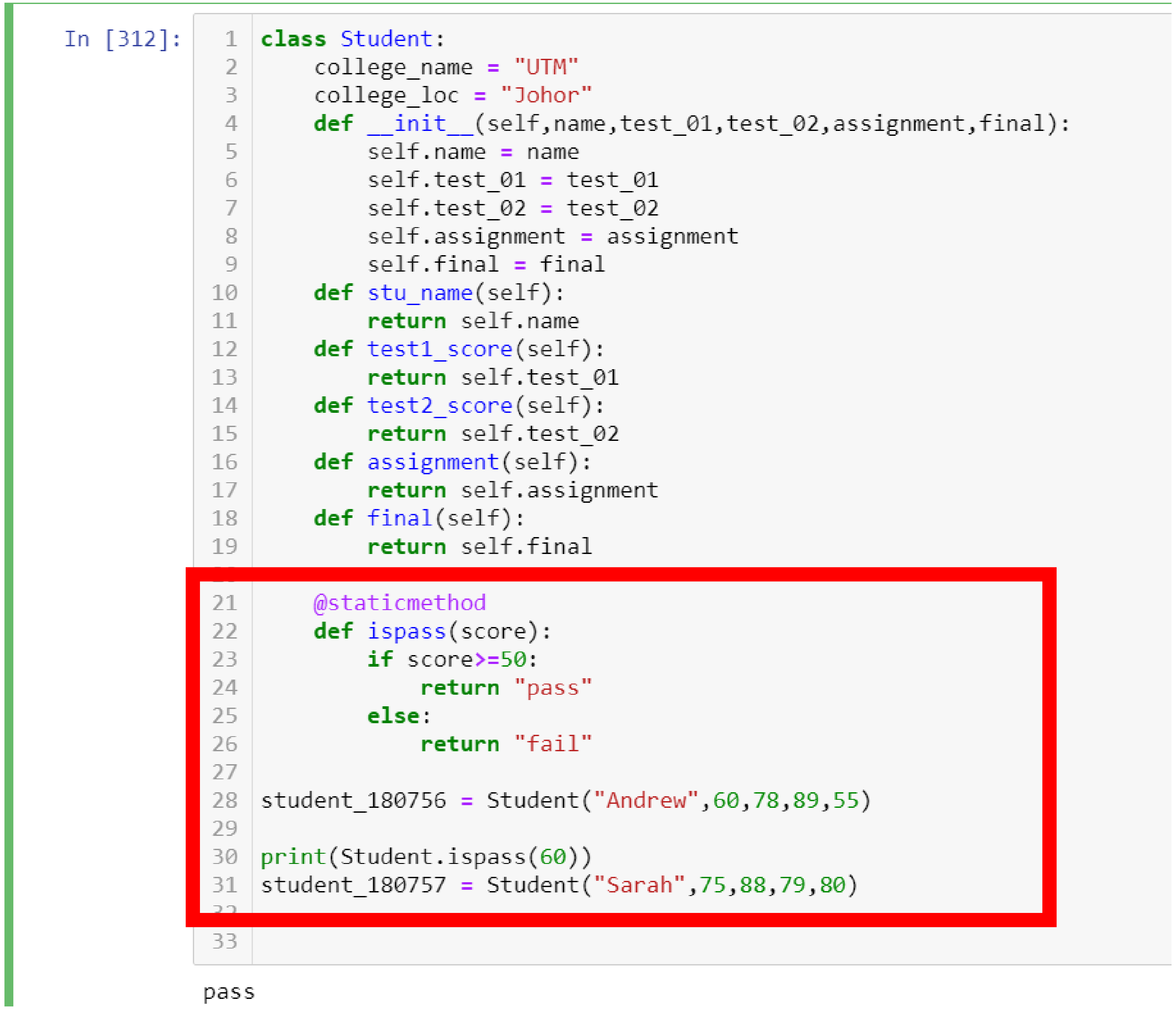
Instance & Class Methods



```python
    def summary(self):
        midterm_score=round((self.test_01+self.test_02)/200*20,1)
        assignment_score=round(self.assignment/100*20,1)
        final_score= round(self.final/100*60,1)
        print("College information is defined in the class method")
        print("-----------------------------------------")
        Student.university()
        print()
        print("Student's information is defined in the instances")
        print("-----------------------------------------")
        print("Student's Name: ",self.name)
        print("Midterm score: ",midterm_score,"% /20%")
        print("Assignment score: ", assignment_score,"% /20%")
        print("Final Test score: ", final_score,"% /60%")
        print("Total score: ",midterm_score+assignment_score+final_score,"% /100%")

student_180756 = Student("Andrew",60,78,89,55)
student_180756.summary()
print("\n===============================================================================\n")
student_180757 = Student("Sarah",75,88,79,80)
student_180757.summary()
```

Static Method

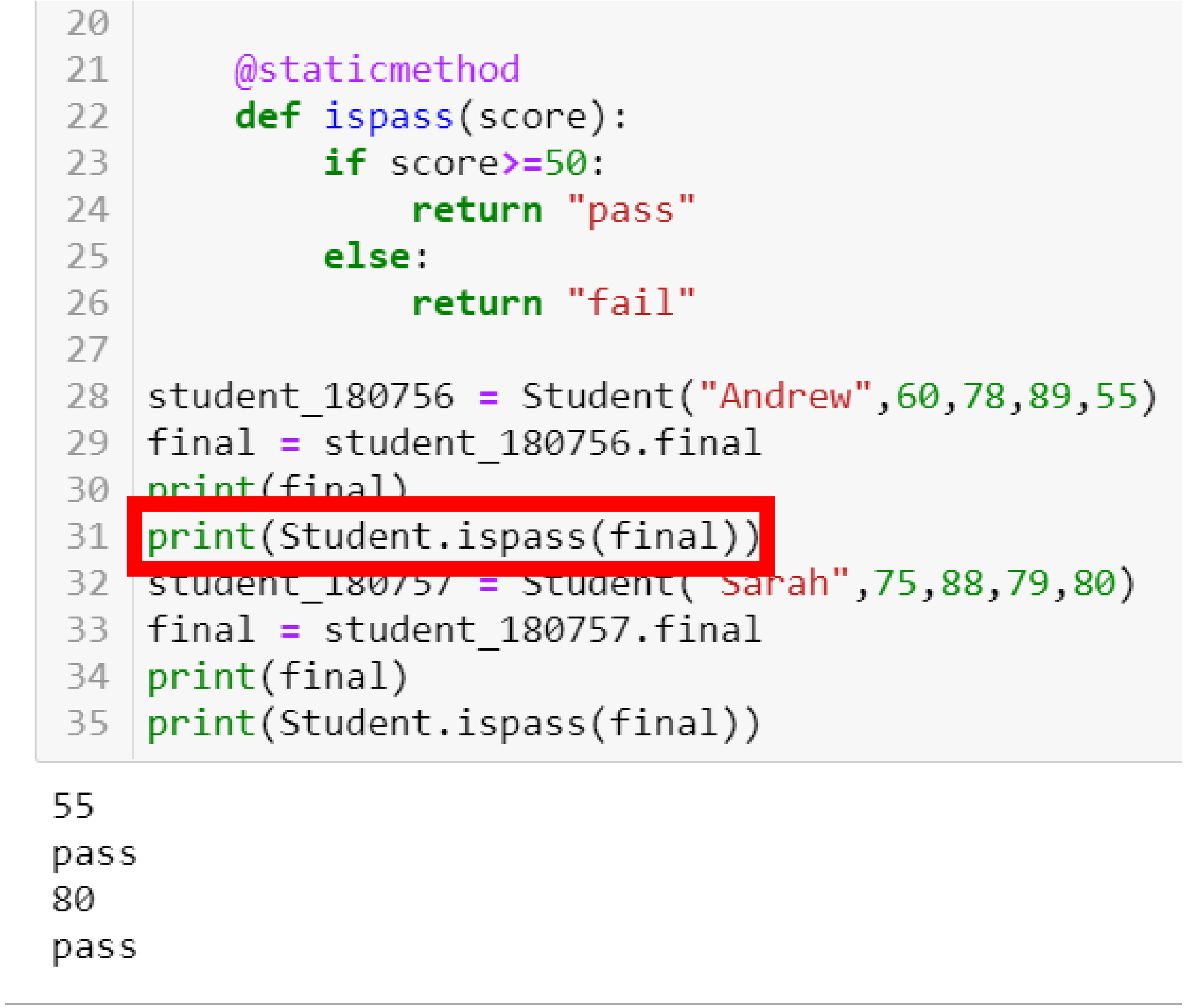
Static Method



In [312]:

```python
class Student:
    college_name = "UTM"
    college_loc = "Johor"
    def __init__(self,name,test_01,test_02,assignment,final):
        self.name = name
        self.test_01 = test_01
        self.test_02 = test_02
        self.assignment = assignment
        self.final = final
    def stu_name(self):
        return self.name
    def test1_score(self):
        return self.test_01
    def test2_score(self):
        return self.test_02
    def assignment(self):
        return self.assignment
    def final(self):
        return self.final

    @staticmethod
    def ispass(score):
        if score>=50:
            return "pass"
        else:
            return "fail"

student_180756 = Student("Andrew",60,78,89,55)

print(Student.ispass(60))
student_180757 = Student("Sarah",75,88,79,80)

```

```
pass
```



```
    @staticmethod
    def ispass(score):
        if score>=50:
            return "pass"
        else:
            return "fail"

student_180756 = Student("Andrew",60,78,89,55)
final = student_180756.final
print(final)
print(Student.ispass(final))
student_180757 = Student("Sarah",75,88,79,80)
final = student_180757.final
print(final)
print(Student.ispass(final))
```

```
55
pass
80
pass
```

Thank you