

Copyright © 2021 Quandatics Academy Sdn Bhd

All rights reserved. All course materials, slides and notes are protected by Quandatics Academy Sdn Bhd. Course materials are copyrighted according to the owner of the content or the principal body. You may take notes on your own, however, you are not allowed to reproduce, distribute, upload or display any of the course materials in any way – whether or not fees is charged – without express written consent from Quandatics Academy Sdn Bhd.

Class



Trainer : Foo Chee Chuan

Class

UTM (Johor)



Students



- **StudentID**
- **Name**

Initialization `__init__(self)`

Identify each students

```
In [252]: 1 class Student:
          2     def __init__(self, name):
          3         self.name = name
          4
          5     def stu_name(self):
          6         return self.name
          7
          8 student_180756 = Student("Andrew")
          9 student_180757 = Student("Sarah")
         10 print(student_180756.stu_name())
         11 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**
- **Name : Andrew**
- **Age : 25**
- **Gender : Male**



- **StudentID : 180757**
- **Name : Sarah**
- **Age : 21**
- **Gender : Female**


```
In [ ]: 1 class Student:
        2     def __init__(self, name, age, gender):
        3         self.name = name
        4         self.age = age
        5         self.gender = gender
        6
        7     def stu_name(self):
        8         return self.name
        9     def stu_age(self):
       10         return self.age
       11     def stu_gender(self):
       12         return self.gender
       13
       14 student_180756 = Student("Andrew", 25, "Male")
       15 student_180757 = Student("Sarah", 21, "Female")
       16
```

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



```
15 student_180756 = Student("Andrew",60,78)
16 print(student_180756.stu_name(),"'s test scores:")
17 print("Test 1 score: ",student_180756.test1_score())
18 print("Test 2 score: ",student_180756.test2_score())
19 print()
```



```
20 student_180757 = Student("Sarah",75,88)
21 print(student_180757.stu_name(),"'s test scores:")
22 print("Test 1 score: ",student_180757.test1_score())
23 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%)**

$$\text{Mid-term} = (\text{test 1} + \text{test 2})/200*20$$

In [267]:

```

1  class Student:
2
3      def __init__(self,name,test_01,test_02):
4          self.name = name
5          self.test_01 = test_01
6          self.test_02 = test_02
7
8      def stu_name(self):
9          return self.name
10     def test1_score(self):
11         return self.test_01
12     def test2_score(self):
13         return self.test_02
14
15     student_180756 = Student("Andrew",60,78)
16     test_01 = student_180756.test1_score()
17     test_02 = student_180756.test2_score()
18     mid_term = round((test_01 + test_02)/200*20,1)
19     print(student_180756.stu_name(),"'s mid-term score is: ",mid_term,"%")
20
21     student_180757 = Student("Sarah",75,88)
22     test_01 = student_180757.test1_score()
23     test_02 = student_180757.test2_score()
24     mid_term = round((test_01 + test_02)/200*20,1)
25     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]: 1 class Student:
2
3     def __init__(self,name,test_01,test_02):
4         self.name = name
5         self.test_01 = test_01
6         self.test_02 = test_02
7
8     def stu_name(self):
9         return self.name
10    def test1_score(self):
11        return self.test_01
12    def test2_score(self):
13        return self.test_02
14    def mid_term(self):
15        return round((self.test_01 + self.test_02)/200*20,1)
16
17 student_180756 = Student("Andrew",60,78)
18 print(student_180756.stu_name(),"'s mid-term score:")
19 print(student_180756.mid_term())
20 student_180757 = Student("Sarah",75,88)
21 print(student_180757.stu_name(),"'s mid-term score:")
22 print(student_180757.mid_term())
```

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

Class Methods

UTM (Johor)



Students



What's common
among the
students?



Class Methods

```
In [284]: 1 class Student:
2
3     college_name = "UTM"
4     college_loc = "Johor"
5
6     def __init__(self, name, test_01, test_02):
7         self.name = name
8         self.test_01 = test_01
9         self.test_02 = test_02
10
11     def stu_name(self):
12         return self.name
13     def test1_score(self):
14         return self.test_01
15     def test2_score(self):
16         return self.test_02
17     def mid_term(self):
18         return round((self.test_01 + self.test_02)/200*20,1)
19
```

```
19
20     @classmethod
21     def university(cls):
22         print("College Name: ", cls.college_name)
23         print("College Location: ", cls.college_loc)
24         print()
25
26
27 Student.university()
28 student_180756 = Student("Andrew", 60, 78)
29 student_180756.university()
30 student_180757 = Student("Andrew", 60, 78)
31 student_180757.university()
32
```

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]: 1 class Student:
2         studentCount = 0
3         def __init__(self, name, test_01, test_02, assignment, final):
4             self.name = name
5             self.test_01 = test_01
6             self.test_02 = test_02
7             self.assignment = assignment
8             self.final = final
9             Student.studentCount += 1
10        def stu_name(self):
11            return self.name
12        def test1_score(self):
13            return self.test_01
14        def test2_score(self):
15            return self.test_02
```

```
20
21         @classmethod
22         def NumberOfStudents(cls):
23             return cls.studentCount
24
25 student_180756 = Student("Andrew", 60, 78, 89, 55)
26 print(Student.NumberOfStudents())
27 student_180757 = Student("Sarah", 75, 88, 79, 80)
28 print(Student.NumberOfStudents())
```

1
2

Instance & Class Methods


```

25     def summary(self):
26         midterm_score=round((self.test_01+self.test_02)/200*20,1)
27         assignment_score=round(self.assignment/100*20,1)
28         final_score= round(self.final/100*60,1)
29         print("College information is defined in the class method")
30         print("-----")
31         Student.university()
32         print()
33         print("Student's information is defined in the instances")
34         print("-----")
35         print("Student's Name: ",self.name)
36         print("Midterm score: ",midterm_score,"% /20%")
37         print("Assignment score: ", assignment_score,"% /20%")
38         print("Final Test score: ", final_score,"% /60%")
39         print("Total score: ",midterm_score+assignment_score+final_score,"% /100%")
40
41 student_180756 = Student("Andrew",60,78,89,55)
42 student_180756.summary()
43 print("\n===== \n")
44 student_180757 = Student("Sarah",75,88,79,80)
45 student_180757.summary()

```

Static Method

Static Method

```
In [312]: 1 class Student:
2         college_name = "UTM"
3         college_loc = "Johor"
4         def __init__(self,name,test_01,test_02,assignment,final):
5             self.name = name
6             self.test_01 = test_01
7             self.test_02 = test_02
8             self.assignment = assignment
9             self.final = final
10        def stu_name(self):
11            return self.name
12        def test1_score(self):
13            return self.test_01
14        def test2_score(self):
15            return self.test_02
16        def assignment(self):
17            return self.assignment
18        def final(self):
19            return self.final
20
21        @staticmethod
22        def ispass(score):
23            if score >= 50:
24                return "pass"
25            else:
26                return "fail"
27
28        student_180756 = Student("Andrew",60,78,89,55)
29
30        print(Student.ispass(60))
31        student_180757 = Student("Sarah",75,88,79,80)
32
33
```

pass

```

20
21     @staticmethod
22     def ispass(score):
23         if score >= 50:
24             return "pass"
25         else:
26             return "fail"
27
28 student_180756 = Student("Andrew", 60, 78, 89, 55)
29 final = student_180756.final
30 print(final)
31 print(Student.ispass(final))
32 student_180757 = Student("Sarah", 75, 88, 79, 80)
33 final = student_180757.final
34 print(final)
35 print(Student.ispass(final))

```

```

55
pass
80
pass

```

Thank you

