This assignment will be evaluated after the deadline passes. You will get your score 48 hrs after the deadline. Until then the score will be shown as Zero.

Common Data for questions 1 to 7 in this assignment

Execute the code-snippet given below and answer all the questions that follow.

```
class Student:
    count = 0
    def __init__(self, name, roll, maths, physics, chemistry):
        Student.count += 1
        self.roll = roll
        self.name = name
        self.maths = maths
        self.physics = physics
        self.chemistry = chemistry

class Group:
    def __init__(self):
        self.members = [ ]

    def add(self, student):
        self.members.append(student)

    def print_members(self):
        for student in self.members:
            print(student.name)
```

1) We wish to create an object named `student` of type `Student`. Select all the correct ways of doing this. [MSQ] ***1 point***

- [ ]
```
student = Student()
```

- [ ]
```
student = Student('Anish')
```

- [x]
```
student = Student('Anish', 4, 90, 95, 100)
```

- [ ]
```
student = Student(self, 'Anish', 4, 90, 95, 100)
```

2) **scores.csv** is a CSV file with the following content:

```
roll,name,maths,physics,chemistry
1,aditya,90,85,78
2,aishwarya,70,89,99
3,anish,90,85,100
4,deeptha,100,95,85
5,lakshmi,95,90,100
```

What is the output of the following snippet of code?

```
Student.count = 0
f = open('scores.csv', 'r')
f.readline()    # ignore the header
students = [ ]
for line in f:
    roll, name, maths, physics, chemistry = line.strip().split(',')
    roll, maths, physics, chemistry = int(roll), int(maths), int(physics), int(chemistry)
    students.append(Student(name, roll, maths, physics, chemistry))
f.close()
print(Student.count)
```

5

2 points

3) Write a method for the `Student` class that returns the sum of the marks scored by the student in all three subjects. ***1 point***

```
def total():
    return maths + physics + chemistry
```

```
def total():
    return self.maths +  self.physics + self.chemistry
```

```
def total(self):
    return maths +  physics + chemistry
```

```
def total(self):
    return self.maths +  self.physics + self.chemistry
```

4) What is the output of the following snippet of code? ***2 points***

```
study_group = Group()
study_group.add(Student('Lathika', 1, 100, 90, 80))
study_group.add(Student('Keerthana', 2, 80, 70, 60))
study_group.add(Student('Sourabh', 3, 100, 50, 60))
study_group.print_members()
```

-
```
Lathika
Sourabh
Keerthana
```

-
```
Sourabh
Keerthana
Lathika
```

-
```
Keerthana
Lathika
Sourabh
```

-
```
Lathika
Keerthana
Sourabh
```

5) Is the following statement true or false? ***2 points***

`print(Group.members)` will print the value of the attribute `members`.

- True
- False

6) Write a method named `remove` for the class `group` that accepts the roll number of a student as argument. It should perform the following function: ***3 points***

If the student is not there in the group, it should print the string "Student not found"
If the student is found in the group, it should remove the student from the group.

```
def remove(self, roll):
```

```
    if roll in self.members:
        self.members.remove(roll)
    else:
        print('Student not found')
```

```
def remove(self, roll):
    found = False
    for student in self.members:
        if student.roll == roll:
            found = True
            break
    if found:
        self.members.remove(roll)
    else:
        print('Student not found')
```

```
def remove(self, roll):
    found = False
    for student in self.members:
        if student.roll == roll:
            found = True
            break
    if found:
        self.members.remove(student)
    else:
        print('Student not found')
```

```
def remove(self, roll):
    found = False
    for student in self.members:
        if student.roll == roll:
            found = True
    if found:
        self.members.remove(student)
    else:
        print('Student not found')
```

7) Consider the following method that is added to the class Group: ***4 points***

```
# Assume that the random library has been imported and is available
def pick_leader(self):
    potential = [ ]
    for student in self.members:
```

```
        avg = (student.maths +
               student.physics +
               student.chemistry) // 3
        if avg >= 80:
            potential.append(student)
    if len(potential) > 0:
        index = random.randint(0, len(potential) - 1)
        self.leader = potential[index].name
    else:
        self.leader = None
```

Consider the following study group.

```
study_group = Group()
study_group.add(Student('Lathika', 1, 80, 80, 80))
study_group.add(Student('Keerthana', 2, 80, 70, 60))
study_group.add(Student('Sourabh', 3, 75, 85, 80))
study_group.add(Student('Nikhil', 4, 100, 79, 59))
study_group.pick_leader()
print(study_group.leader)
```

Based on your understanding of the method `pick_leader`, which of the following students could become a leader of this group?

- [x] `Lathika`
- [ ] `Keerthana`
- [x] `Sourabh`
- [ ] `Nikhil`

8) What is the output of the following code-snippet? ***2 points***

```
D = {'Anita': 23, 'Ashwin': 43,
     'Ahana': '24',
     'Adarsh': 30, 'Archana': 15}
try:
    # iterates through the keys from left to right
    for key in D:
        value = D[key]
        if type(value) == str:
            raise 'Error'
```

```
        print(f'{key}:{value}')
except:
    print("Values cannot be strings")
```

```
Anita:23
Ashwin:43
Ahana:24
Adarsh:30
Archna:15
```

```
Anita:23
Ashwin:43
```

```
Values cannot be strings
```

```
Anita:23
Ashwin:43
```

```
Anita:23
Ashwin:43
Values cannot be strings
```

9) What is the output of the following snippet of code? ***2 points***

```
L = [1, 3, -1, 4, -2, 5, 3]

try:
    n = 10
    for i in range(n):
        if L[i] < 0:
            L[i] = 0
except IndexError:
    for i in range(n - len(L)):
        L.append(0)
finally:
    print(L)
```

```
[1, 3, -1, 4, -2, 5, 3]
```

```
[1, 3, 0, 4, 0, 5, 3]
```

```
[1, 3, 0, 4, 0, 5, 3, 0, 0, 0]
```

This code doesn't print anything to the console.

10) What is the output of the following snippet of code? ***1 point***

```
try:
    L = [x for x in range(10)]
    f = open('numbers.txt', 'w')
    for x in L:
        f.write(x)
except FileNotFoundError:
    print('File was not found')
except:
    print('This is some other error')
finally:
    print('The file has been closed')
    f.close()
```

```
File was not found
```

```
File was not found
The file has been closed
```

```
This is some other error
```

```
This is some other error
The file has been closed
```