Knowledge Quiz 26: Overloaded Operators

Due Nov 20 at 1pm **Points** 5 **Questions** 2 **Available** until Dec 4 at 11:59pm **Time Limit** 60 Minutes

Instructions

Prior to completing this quiz, be sure to read:

- Sections 8.4: Overloaded Operators (p. 256-264)

Please also go over Practice Problems 8.6 and 8.7 in the textbook (solutions at the end of the chapter) AND Learning Quiz 26 before attempting this quiz.

You may attempt this quiz ONE time, and you have 60 minutes to complete this quiz. Only submissions score *prior to the deadline* will count towards the final course grade. No late submissions will be accepted.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	24 minutes	1 out of 5 *

* Some questions not yet graded

ⓘ Correct answers are hidden.

Score for this quiz: **1** out of 5 *
Submitted Nov 21 at 2:01pm
This attempt took 24 minutes.

Partial

Question 1

1 / 2 pts

Suppose we have a Team class as initialized below. We would like to be able to use the ==, !=, >, >=, <, and <= operators to compare the team sizes. Fill in the blanks below so that the operator will logically return True/False based on the team sizes:

```
class Team:
    'Represents a team'

    def __init__(self, count):
        'initializes the team and team count'
        self.count = count

    def getcount(self):
        'returns the number of team members'
        return self.count
```

```
def __ eq __(self, other):
    'returns True if the team counts are equal'
    self.count == other.count
def __ ne __(self, other):
    'returns True if the team counts are not equal'
    self.count != other.count
def __ ne __(self, other):
    'returns True if current team count is greater than or equal to other team count'
    self.count >= other.count
def __ ge __(self, other):
    'returns True if current team count is greater than other team count'
    self.count > other.count
def __ gt __(self, other):
    'returns True if current team count is less than or equal to other team count'
    self.count <= other.count
def __ lt __(self, other):
    'returns True if current team count is less than other team count'
    self.count < other.count
```

Answer 1:

eq

Answer 2:

ne

Answer 3:

ne

Answer 4:

ge

Answer 5:

gt

Answer 6:

lt

Question 2

Not yet graded / 3 pts

Suppose we have a Team class as initialized below. We would like to be able to use the + and - operators to state the total or difference in number of team members across the two teams.

```
class Team:
    'Represents a team'

    def __init__(self, count):
        'initializes the team and team count'
        self.count = count

    def getcount(self):
        'returns the number of team members'
        return self.count
```

Add to the class above so that a programmer using the Team class can get an output as follows:

```
>>> team1 = Team(10)
>>> team2 = Team(15)
>>> print(team1 + team2)
25
>>> print(team1 - team2)
-5
>>> print(team1 * 2)
20
```

Your Answer:

```
class Team:
    'Represents a team'

    def __init__(self, count):
        'initializes the team and team count'
        self.count = count

    def getcount(self):
        'returns the number of team members'
        return self.count

    def __add__(self, other):          ## + operator
        ''' return the two teams added '''
        return self.getcount() + other.getcount()

    def __sub__(self, other):          ## - operator
        ''' return the two teams added '''
        return self.getcount() - other.getcount()

    def __mul__(self, other):          ## * operator
        ''' return the two teams added '''
        return self.getcount() * other
```

Quiz Score: **1** out of 5