Week-10, Graded, Theory

Week-10, Graded, Theory

- Problem 1
  - Question 1
  - Answer
  - Solution
  - Question 2
  - Answer
  - Solution
  - Question 3
  - Answer
  - Solution
  - Question 4
  - Answer
  - Solution
  - Question 5
  - Answer
  - Solution
  - Question 6
  - Answer
  - Solution
  - Question 7
  - Answer
  - Solution
  - Question 8
  - Answer
  - Solution
  - Question 9
  - Answer
  - Solution
  - Question 10
  - Answer
  - Solution

Problem 1

Common data for the following questions.

Execute the code-snippet given below and answer the questions that follow. This code-snippet will be referred to as `main-code` wherever required.

```
class Student:
    count = 0
    def __init__(self, name, rollNo=None, mathMarks=None, physicsMarks=None, chemistryMarks=None):
        Student.count += 1
        self.rollNo = rollNo
        self.name = name
        self.mathMarks = mathMarks
        self.physicsMarks = physicsMarks
        self.chemistryMarks = chemistryMarks

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

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

    def remove(self, studentObj):
        self.members.remove(studentObj)

    def printMembers(self):
        for i in self.members:
            print(i.name)
```

Question 1

What is(are) the required parameters to create a `Student` object? [MSQ]

(a) `self`
(b) `rollNo`
(c) `name`
(d) `mathMarks`
(e) `physicsMarks`
(f) `chmistryMarks`

Answer

(c) `name`

Solution

Except `name`, all other parameters have default value `None`. Hence, `name` is the only required parameter. `self` is not required as parameter in creating an object of a class.

Question 2

What does the `count` attribute represent?

(a) Number of classes
(b) Number of objects
(c) Number of `Student` objects
(d) No meaningful information

Answer

(c) Number of `Student` objects

Solution

`count` is the attribute of class, the value of `count` is initialized to `0` in the beginning. The value of `count` is incremented by `1` once call of `__init__` function of `Student` class. The `__init__` function is called only during the object creation, hence `count` represents of the number of `Student` objects.

Question 3

The following code is executed after executing `main-code`. What will be the value of `count`? [NAT]

```
s0 = Student('Bhuvanesh', 0, 68, 64, 78)
s1 = Student('Harish', 1, 62, 45, 91)
```

Answer

2

Solution

The value of `count` is incremented by `1` for every new object. Two objects have been created then the value of `count` is `2`.

Question 4

The following code-snippet is executed subsequent to the execution of the code given in the previous question. Will this throw any error?

```
s30 = Student('Nirmala', mathMarks=89, physicsMarks = 77, chemistryMarks = 67)
```

(a) Yes, it will throw an error
(b) No, it will not throw an error

Answer

(b)

Solution

The above will not throw an error because required parameter `name` is given and `Student` is a user defined class.

Question 5

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

(a)

```
def total():
    if mathMarks != None and physicsMarks != None and chemistryMarks != None:
        return mathMarks +  physicsMarks + chemistryMarks
```

(b)

```
def total():
    if (self.mathMarks != None and
        self.physicsMarks != None and
        self.chemistryMarks != None):
        return self.mathMarks +  self.physicsMarks + self.chemistryMarks
```

(c)

```
def total(self):
    if (mathMarks != None and
        physicsMarks != None and
        chemistryMarks != None):
        return mathMarks +  physicsMarks + chemistryMarks
```

(d)

```
def total(self):
    if (self.mathMarks != None and
        self.physicsMarks != None and
        self.chemistryMarks != None):
        return self.mathMarks +  self.physicsMarks + self.chemistryMarks
```

Answer

(d)

Solution

A method of a class requires `self` as first parameter. Every attribute of object should accessed using `self`. For an example the physics marks of the object is accessed by `self.physicsMarks`.

Question 6

The following code is executed subsequent to the execution of the code in the previous question.

```
yoga = Group()
yoga.add(s0)
yoga.add(s1)
```

Is the following statement true or false.

> The list `members` will be empty at the end of execution.

(a) True
(b) False

Answer

(b) False

Solution

`yoga.add(s0)` adds the object `s0` into the list `yoga.members` similarly for `yoga.add(s1)` as well. Thus, `members` is not empty.

Question 7

Is the following statement true or false?

> `print(Group.members)` will prints the list `members`.

(a) True
(b) False

Answer

(b) False

Solution

The list `members` is a the objects attribute not a class attribute.

Question 8

The following code is executed after executing the code in the previous question.

```
s2 = Student('Shashank', 2, 57, 54, 77)
s3 = Student('Rida', 3, 42, 53, 78)
s4 = Student('Ritika', 4, 87, 64, 89)
s5 = Student('Akshaya', 5, 71, 92, 84)

yoga.add(s3)
yoga.add(s4)

dance = Group()
dance.add(s2)
dance.add(s0)
dance.add(s4)
```

What does `yoga.printMembers()` prints?
(a)

```
Ritika
```

(b)

```
Ritika
Rida
Harish
Bhuvanesh
```

(c)

```
Bhuvanesh
Harish
Rida
Ritika
```

(d) Nothing will be printed

Answer

(c)

Solution

`yoga.printMembers()` will print the name of the `Student` objects present in the list `yoga.members`. `yoga.members` be `[s0, s1, s2, s3]`. Hence, the name of the object will be printed in the same order.

Question 9

What does `dance.printMembers()` print?

(a)

```
Shashank
Bhuvanesh
Ritika
```

(b)

```
Ritika
```

(c)

```
Bhuvanesh
Harish
Rida
Ritika
```

(d) Nothing will be printed

Answer

(a)

Solution

Self explanatory based on previous question's solution.

Question 10

What do the elements in the list `members` represent?

(a) Names of the students
(b) Roll number of the student
(c) `Student` class objects
(d) Object names in strings

Answer

(c) `Student` class objects

Solution

The objects of `Student` class are added into the object attribute `members` using `add` method of `Group` class.