Fundamentals of Programming for Business

Lab Assignment 5

Due Date: Sunday, Apr. 25th, 2021 by 11:59 pm

Important: *Submit your lab assignment as a single file via Canvas named LastName_FirstName_Lab_LabNumber.zip (e.g., Smith_John_Lab_05.zip). No late submissions will be accepted.*

Description:

In this lab assignment, you will use object-oriented programming technique to model a basic college class that contains some students.

Instructions:

You will need to implement the following Python classes: **Class** and **Student**.

Student includes the following data attributes:

- First name
- Last name
- Gender
- Student id
- email

You may provide additional attributes as needed.

Your Python class must also include some accessor (getter) and mutator (setter) methods to return and manipulate the above data items. The Student class must contain the following methods:

- Methods to get and set the following data attributes:
  - First name
  - Last name
  - Gender
  - Student id
  - Email
- A method to get the student's full name
- The special method __str__ to return a string representation of a student object. Use the following template to print the information:

{student full name}, has the following email address: **{student email}**. **{His/Her}** id is: **{student id}**

Class includes the following attributes:

- Associated term
- Section number
- Subject
- Course number
- CRN
- Title
- Credit hours
- List of students
- A list of grades for each student
- Instructor name

Provide the following methods in the implementation of **Class**:

- Methods to get and set the following attributes:
    - Associated term
    - Section number
    - Subject
    - Course number
    - CRN
    - Title
    - Credit hours
    - Instructor name

- A method to add a single student: addStudent()
- A method to add a list of students: addStudents()
- A method to add a single grade for a specific student: addGrade()
- A method to add multiple grades (a list of grades) for a specific student: addGrades()
- A method to return the grades for a specific student: getGrades()
- A method to return the list of current students: getStudentsList()
    - Return the name and id for each student
- A method to return the number of students: getStudentsNum()
- A method to calculate and return the average score for a single student: getAverage()
- A method to calculate and return the class average score: getClassAverage()

To test your program, use the following code:

```
python = Class('Fall 2021', '01', 'IT', 2114, 1234,
               'Fundamentals of programming', 3)

student1 = Student('John', 'Smith', 'male', 123, 'john.smith@gmail.com')
student2 = Student('Sara', 'Hamid', 'female', 124, 'sara.hamid@gmail.com')
student3 = Student('Lu', 'Zhang', 'male', 125, 'lu.zhang@gmail.com')

python.addStudents([student1, student2])
python.addStudent(student3)

print(student1)
print(student2)

python.addGrade(student1, 100)
python.addGrades(student1, [95, 80])

python.addGrades(student2, [45, 45, 45, 50])
python.addGrades(student3, [89, 79, 66, 80])

python.addGrade(student1, 95)

print(python.getGrades(student1))

print(python.getAverage(student1))
print(python.getClassAverage())

python.getStudentsList()
python.getStudentsNum()
```

Expected results:

John Smith, has the following email address: john.smith@gmail.com. His id is: 123

Sara Hamid, has the following email address: sara.hamid@gmail.com. Her id is: 124

[100, 95, 80, 95]

John Smith has a score average of: 92.5

The class average is: 72.42

John Smith - id: 123

Sara Hamid - id: 124

Lu Zhang - id: 125

There are 3 students in this class.

Grading:

Each assignment is out of 60 points, graded on:

- Does the code work properly? If doesn't work, describe where you had problems.
- Is the code well commented and readable?
- Did you implement all the required parts?

Submission:

1. For each assignment, create a new directory on your computer, and name it: LastName_FirstName_Lab_LabNumber.zip (Ex: Smith_John_Lab_05.zip).
2. Copy all your Python scripts (the final version of your files) to your directory. You can also add a README.txt that contains information on how to run your code.
3. Zip your folder using the same name as your directory.
4. Go to the Canvas course site. Click the Assignment link for the assignment you wish to submit. You will then see a screen allowing you to submit your assignment.
5. The "Choose File" button allows you to select the file you wish to turn in.
6. Use the Submit button to submit your file for grading.
7. After using Submit, WAIT until you see a confirmation screen showing that your assignment was successfully submitted. If you close the browser window before receiving this confirmation, your submitted file may be corrupted.
8. **No files sent via email will be accepted or considered**. No other submissions, of any form other than that described in this handout, will be accepted or considered.