Assignment 2 – CPS 109

In this assignment, you are going to write a program for processing class lists. The program reads the list of individuals from a text file, performs a set of operations and writes the output to a text file. This assignment helps you practice OOP using Python.

The input file format

The input file is a text file containing a class list. The first 4 lines contain information about the class. Starting from the 5th line, each line contains information about a person. Each line has the following format:

```
Last Name, First Name Username ID Role
```

where all the fields are strings. The Username is the first part, i.e., up to @, of the email address. The Role can be one of Student, TA, or Instructor. An example of a line is

```
Doliskani, Jake javad.doliskani RU01157776 Instructor
```

The program

Your program should contain the following classes, which are all going to be in one file called assignment2.py.

Person class.
Instance variables

- `first_name, last_name, username, id_num`

Methods

- `__init__`: accepts the above instance variables
- `__str__`: returns a string of the format:

  ```
  First name: first_name
  Last name: last_name
  Username: username
  ID: id_num
  ```

  Note that each entry is in one line.

Student class.
Subclass of Person.

Instructor class.
Subclass of Person.

TeachingAssistant class.
Subclass of Person.

Parser class.
Instance variables

- `students`: which is a list of students
- `instructors`: which is a list of instructors
- `tas`: which is a list of teaching assistants

Methods

- `__init__`: initializes the above lists to empty lists.
- `parse`: accepts a filename as a string. This methods opens the file in text mode, reads the records in the file and populates the lists of students, instructors and TA's. Based on the role column in each line, a different class should be instantiated and added to the corresponding list. For example, if the role is Student, an instance of the Student class is created and added to the students list.
- `get_students`: returns the list of students
- `get_instructors`: returns the list of instructors
- `get_tas`: returns the list of TA's

Main class.
Instance variables

- `parser`: an instance of the class Parsser.

Methods

- `__init__`: initializes the `parser` instance variable.
- `parse_file`: accepts a string `filename` which is the name of a text file. This method uses the `parser` to parse the file with name `filename`.
- `get_students`: accepts a string `str1`. Returns a list which contains all the students from `parser.get_students()` whose `id_num` contains `str1`.
- `write_to_file`: accepts two arguments, `persons` and `filename`. The argument `persons` is a list of persons. The argument `filename` is a string which is the name of file. This method writes the list `persons` to the file with name `filename`. The format of the output file should be as follows:

```
First name: first_name1
Last name: last_name1
Username: username1
ID: id_num1

First name: first_name2
Last name: last_name2
Username: username2
ID: id_num2

...
```

Note that the `__str__` method in person gives you each of these groups of lines. You need to add newline characters where needed.

Testing

You can test your program with the test.py file. Open a terminal and run
`python test.py`
This will use the input file called class_list, and create a file called new_list which contains the list of students whose id_num contains "91".

Submission

You only need to submit assignment2.py to D2L. DO NOT copy and paste the code into D2L, upload the file!

Plagiarism detection

You are to work alone when writing your code. You can discuss general ideas with your classmates, but you cannot copy code or develop code together nor take code from the web. We will be using the Measure of Software Similarity (MOSS) to identify cases of possible plagiarism; see the following link for details: http://theory.stanford.edu/~aiken/moss. Note, MOSS can detect changing identifiers and rearranging code. The Department of Computer Science takes the act of plagiarism very seriously. Those caught plagiarizing (both originators and copiers) will be sanctioned. Please see Ryerson University's Policy 60 for possible penalties and consequences:
http://ryerson.ca/senate/policies/pol60_procedures.pdf.