CSE111 Section 3
Project (Optional)
Summer 2023
Instructor: Dibyo Fabian Dofadar

- -

Project Description:

Suppose, you are hired to design a class called "**BracPepol**". To complete the implementation, you have to follow each and every requirement below properly:

1. **Read from the Data:**
   You have the information of all the students in this file (**click here**). You can see, there is information about 5000 students. Apart from the header (first row), each row will be considered as a student object. You have to read the .csv file in your implementation and create the objects. You can use loops instead of writing 5000 lines of driver code to create these objects.

2. **Constructor Design:**
   - While creating an object, you have to pass all the information of each row from the provided data to the constructor. So, the constructor will take the following parameters:
     - ***name, gender, location, birthdate, blood_group, phone, department, enrolled_year, completed_credits, current_cgpa***
   - Inside the constructor, you also have to generate a unique ID. The format is:
     Last two digits of enrolled year + one digit department code + four digits serial
     You can use the following dictionary to access the department with their corresponding department code.
     departmentCodeDict = {'Archi': 1, 'CSE': 2, 'ESS': 3, 'EEE': 4, 'ENH': 5,
     'MNS': 6, 'Pharma': 7, 'BBS': 8, 'Law': 9}
     For example, in the first entry of your data, you can see a student named "Samuel Whitaker" enrolled in 2016 (last two digits are "**16**") and his department is BBS (department code "**8**"). As he is the first entry of the data, his four digit serial number will be "**0001**". So, his **7-digit** unique ID will be "1680001".
   - You can additionally create and use as many instance or class variables you need.

3. **Class Methods Design:**
   You have to create and invoke the following class methods:
   - ***male_femaleRatio(cls)***:
     - This class method will print the ratio between male and female students.
   - ***siblingFromAnotherMother(cls, year, month = None)***:
     - This class method will print the name of the students who share the same birth year passed as value. [if the month information is not available]
     - Additionally, if the value of month is provided, then it will print the name of the students who share the same birth year and the same month.

- ***availableBloodDonorByLocation(cls, blood_group, location, count = 10)***:
  - This class method will print the name and phone number of the students based on the matches with the blood group and the location.
  - By default, the maximum donor count is 10. That means if the value is not passed, information of the first 10 donors will be displayed.
- ***availableBloodDonorByDept(cls, blood_group, dept = None, count = 10)***:
  - This class method will print the name, phone number and address of the students based on the matches with the blood group and the department.
  - If the value of the department is not passed, information of the first 25 possible donors among all the students will be displayed.
  - By default, the maximum donor count is 10. That means if the value is not passed, information of the first 10 donors in the department will be displayed.
- ***generateProbationStudentInfo(cls, dept = "CSE")***:
  - This class method will print the name, id, department name, phone number and current CGPA of all the students based on the provided department (default "CSE") whose CGPA is less than 2.00 .
- ***findValedictorianCandidates(cls)***:
  - This class method will print the name, id, department and current CGPA of Top 5 highest CGPA holders among all the departments who have completed at least 130 credits.
  - If there are several candidates with the same CGPA, then the student(s) with higher completed credits will get priority. If that becomes the same, then the student who enrolled earlier will be prioritized.
- ***findGoldMedalistCandidates(cls)***:
  - For each department, there will be only one gold medalist candidate who is the highest CGPA holder in that department and has already completed at least 130 credits.
  - This class method will print the name, id and current CGPA (department wise) of possible Gold Medalists Candidates. In your data, you can see there are 9 departments. So, this class method should print at most 9 different lines.
  - If there are several candidates with the same CGPA, then the student with higher completed credits will get priority. If that becomes the same, then the student who enrolled earlier will be prioritized.
- ***findFemaleCoderChampionshipCandidates(cls)***:
  - This class method will print the name and id of maximum 10 female CSE students having a CGPA more than 3.5 and completing at least 30 credits. While finding the candidates, make sure the junior female CSE students

will get more priority. That means students with higher completed credits will be prioritized less.

- ***findSIM_users(cls, operatorName)****:*
  - This class method will print the name, location and phone number of students who use the same operator's SIM card.
  - In the parameter *operatorName*, there are only 4 possible values. You can use the following table for categorizing the users.

SIM Operator Name	Digit After "+8801"
Grameenphone	**3 or 7**
Banglalink	**4 or 9**
TeleTalk	**5**
Robi	**6 or 8**

Instructions:

- This is an **optional** task. You can skip it if you want. But I would highly recommend you to complete it. After successful completion of the project, you will get a chance to recover some of your lost marks from **Quiz and Attendance** marks. **Project completion has no relation with Midterm or Final marks.**
- You have to form a group of 1/2/3 member(s). **More than 3 persons in each group is not allowed.** It is recommended that every group member should contribute almost equally. Do not put too much pressure on any particular group member.
- For file reading, you have to use Python's Built-in **CSV** Library (***import csv***). **You are not allowed to use "Pandas" (or equivalent) Library.**

  The following links might be helpful:
    - CSV File Reading and Writing — Python 3.11.3 documentation
    - Reading and Writing CSV Files in Python
    - How to Read and Write With CSV Files in Python
- You have to submit your group information along with the code file in a Google Form. In the form, you have to choose a Viva/Presentation slot. That would be an informal presentation in your preferred language (Bengali/English/Hindi is recommended for my better understanding). In the presentation, you have to explain your work to me. I will ask a few questions if needed. The total duration of each presentation shouldn't be more than 15 minutes.
- Google Form Link: **click here**
- **Any type of plagiarism will not https://forms.gle/RwPKXJDQZz61J4nu6be tolerated.**
- **After 28th August, no submission will be accepted and no presentation will be taken.**
- Feel free to ask any question in Slack.