CSIT 552 HW1
Topic: Python Basics

Dr. Boxiang Dong

1 Problem Description

Instructions. Please write code in a Python notebook to complete the following tasks.

- Task 1 (10 pts). Define a *Student* class with the following attributes:
  - name: str
  - cwid: str (like your MSU id, it starts with letter M and has 8 digits after it)
  - grades: a list of letters that represents the letter-based grade, (A, B, C, D, and F)
  - gpa: float

  The constructor takes the name as argument and generates cwid and grades by calling the functions implemented in Task 2 - 4.

- Task 2 (10 pts). Implement a private function named *generate_cwid* that generates a random cwid. Note that the first digit in a cwid should not be 0. (Hint: use *itertools.combinations_with_replacement()*).

- Task 3 (10 pts). Implement a private *generator* function named *generate_grade* that returns a random grade in the set of (A, B, C, D, and F).

- Task 4 (10 pts). Implement a private function named *generate_grades* that returns a list of letter-based grades. The list can have more than 10 grades, but it should include 10 non-F grades. This simulates the situation where a student fails a course, he/she will need to take one more course to make up the credits.

- Task 5 (10 pts). Write a function named *calculate_gpa* to update the student's gpa based on the course grades. Assuming each course is 3 credits. The schema to translate letter-based grade to numeric grade is
  - A: 4.0

  - B: 3.0
  - C: 2.0
  - D: 1.0
  - F: 0.0

- Task 6 (10 pts). Implement the magic str function to return the string representation of a student in the format of *name (cwid): gpa*.

- Task 7 (20 pts). Outside the Student class, implement a function named *simulate_students* that

  - Generate 10 students (you are free to provide the names)
  - Print all students based on their GPA from the highest to the lowest
  - If any student has a GPA lower than 2.0 or has failed more than 2 courses, raise an exception where the error message includes the number of students that fall into this case, as well as a list of their cwid.

  Call the function to show the output and exception if any.

- Task 8 (10 pts). Provide type hint for each function.

- Task 9 (10 pts). Provide docstring for each class and function.

2 Submission Guideline

1. Work individually.
2. Please submit a .ipynb file.
3. Submit your solution on Canvas on time. A late penalty of 10 points for each late day applies. Any late for more than three days receives zero automatically.