CMSC 124 – Design and Implementation of Programming Languages
Exercise 3 and 4 – Student Records Using Perl

Create a student record program that takes up to 10 student records. Each student must have the following information:

- Name
- Student Number
- Birthdate (YYYY-MM-DD)
- Contact Number
- Email Address
- Subjects; For each subject:
  - Course Number
  - Course Credit (Units)
  - Grade

The user can do the following operations:

- Create student record
  - Ask for all the aforementioned information
  - Student numbers must be unique, the rest may repeat
- Edit student record
  - Identify the student to be edited using his student number
  - Edit only the information that the use wants to change (don't change information that the user doesn't want to)
- View student records
  - View one students
    - Identify the student to be viewed using his student number
    - Show all aforementioned information, plus his current GWA, computed as

$$GWA = \frac{\sum units \times grade}{\sum units}$$

  - View all students
    - View all students and show all aforementioned information, plus his GWA (computation specified above)
- Search for students using
  - Name
  - Student number
  - Birthdate
  - In all three cases, either full or partial information may be given. The user may also give regex as input if s/he has sufficient knowledge about it.
- Delete student record
  - Identify the student record to be deleted using the student's student number
  - Remove the entry from the list of student records

All data must be entered in a list (students) of hashes (student information). The subjects should be stored in a list within the student information hash, and each subject should be stored in a hash within that list For example, a single student entry may be:

```
%student[0] = (
        “name” => “Kei Peralta”,
        “studentNumber” => “XXXX-XXXXX”,
        “birthdate” => “YYYY-MM-DD”,
        “contactNumber” => “09XXXXXXXXX”,
        “emailAddress” => cnmperalta@gmail.com
        “subjects” => {
                (“courseNumber” => “CMSC 124”,
                 “courseCredit” => 3.0,
                 “courseGrade” => 1.25
                ),
                (“courseNumber” => “CMSC 150”,
                 “courseCredit” => 3.0,
                 “courseGrade” => 2.00
                )
        }
)
```

And, the above entry is just one element of the overall list of student records.

Each operation (create, edit, view, etc.) must have **at least one function**; make your programs as modular as possible.

All inputs must be validated before they are stored in the student record:

- Name – Consists of a first name of at least one up to 30 characters and a last name of at least one up to 30 characters as well.
- Student Number – Four digits followed by a – followed by 5 digits, ex. 2012-12345.
- Contact Number – Either starts with a 0 or +63 followed by a 9 and followed by 9 digits.
- Email Address – Username must have at least 6 characters and at most 30 characters, starts with an alphabetical character (a-z or A-Z), followed by any word character including periods (.), followed by the @ symbol and any legitimate domain name, e.g., .com, .edu.ph, .org, .co.uk, etc.
- Course Number – Three to four upper-case characters, followed by a space, followed by one to three digits; the maximum hundreds value of a course number is 4 (ex., CMSC 400, Ph. D. dissertation)
- Course Credit – Floating number with 2 decimal places.
- Grade – Follow the grading scheme of UP.

BONUS: Sort the students by name. (+1 / 15).
Save to and read from a file containing all student record data (+2 / 15).
Allow users to search all students of a certain course number (+1 / 15).

Good luck!