Program One
Due November 14, 2023

Your assignment is to create a Ruby program that will use data from a single file with student id/names, course numbers/names and enrollment information, and then summarize the input. Failure to follow these directions may affect your grade on this assignment.

Input

- The single input file will be in three parts. Each part will be separated by a blank (empty) line. Your program must not assume any maximum number of students, courses, or enrollment lines. Strings are whitespace separated; this may include ANY combination of spaces or tab characters. Do not assume any specific separator.
- Part 1 Student ids and name
  - There will be an arbitrary number of student info lines in this part. Each line will contain the following:
    - The first string will be the student id. Do not assume a particular number of characters for the id. Assume there will be no spaces in the id.
    - All following strings will constitute the student's full name. Do not assume any order within a name (e.g., first name first, last name first, etc.) Preserve the names exactly as they are, including spaces.
    - The student name may have any number of words, spaces and punctuation.
  - Part 1 ends with one blank line (no spaces or tab on the line).
- Part 2 Course CRNs and names
  - There will be an arbitrary number of course info lines in this part. Each line will contain the following:
    - The first string will be the course CRN. Do not assume a particular number of characters for the CRN. Assume there will be no spaces in the CRN.
    - The rest of the line will be the complete course name. The name may contain a department, number and/or multi-word name. Preserve the complete name exactly as it is given; you may convert whitespace to a single space.
  - Part 2 ends with one blank line (no spaces or tab on the line).
- Part 3 Student ids and CRNs to represent that a student is enrolled in that course. Each line contains exactly two strings.
  - There will be an arbitrary number of student registration info lines in this part. Each line will contain the following:
    - The first string will be the student id. If the student id was not previously read in, output an error message, but continue processing
    - The second string will be the course CRN. If the CRN was not previously read in, output an error message, but continue processing.
  - There will be no other content on these lines.
- Here is a sample input file. This will not be the one used for testing your program. You should test your program with more than one set of test data.

```
A12345     Joseph P. Allen, Jr.
A23456     Brenda Gomez
A34567     Stephanie McGraw

987  CS 424      Programming Languages
123  CS 490      Operating Systems
456  CS 317      Algorithms

A12345     987
A34567     123
A23456     456
A12345     123
```

Output

Output will be to standard output (e.g., using **puts**). Output must include at least these two sections:

- A list of each student with the full course name(s) for which they are enrolled.
  - Optional: the count for the number of courses each student is taking.
- A list of each course (course number and name) with the full list of student name(s) enrolled in that course.
  - Optional: the count for the number of students in each course.

Your program may display additional information as long as the above output is clearly displayed.

Design

Your program will read its input directly from a single text file named “register.txt”. Your program must read directly from a file with this exact name (all lower case) in the same folder as your program. Do not query the user for the filename or anything else. Do not include any path or folder information in your program. The file must be in the same folder as your program.

Your program must create at least two classes, one class for student information and the other for course information. Each class must have at least two methods in addition to its initializer. You may find it useful to create an additional “Seat” class for representing the enrollment.

You must read the input file exactly once.

You must include a header in the program source identifying the assignment/course/instructor, yourself, and the date. See the Submission Guidelines on the Syllabus for additional help.

Submission

Submit exactly one source code file. Your file must be named Ruby-YourName.rb (use your real last name for “YourName”) – do NOT put any spaces in your file name. Follow the Submission Guidelines for programs as found in the syllabus. Follow these additional instructions:

- Include comments at the start of your source code summarizing your overall design, suitable for a person who does not know what the program is about.
- Include comments at the end of your source code giving a summary of what you learned from the assignment and your impressions of Ruby as a programming language.
- Make sure you submit your file as an .rb (text) file, NOT a Word or PDF document.
- Do not submit your test data file(s).
- Do not submit a copy of your output. The instructor will run your program with his/her own data.
- Do not submit any other files. Explanations or comments by you should be included in the source code as Ruby comments.