Programming Evaluation

Overall Expectations Being Evaluated:

A2. describe and use modular programming concepts and principles in the creation of computer programs

A3. design and write algorithms and subprograms to solve a variety of problems;

C1. demonstrate the ability to apply modular design concepts in computer programs;

The Program You are To Create

Use your knowledge of object lists to create a simple database that will make use of the data in **`lifeLostbyCauseData.txt`**. The data represents Canadian potential years of life lost broken down by cause of death, year, sex, and geographical area.

Each record in the data file is composed of six comma-separated fields.

i. Reference Date
ii. Geographical Area (either Canada as a whole or by province)
iii. Sex (Males, Females, or both sexes)
iv. Cause
v. Unit
vi. Value (potential years of life lost)

1. Read the whole input data file into your program, storing each record as a separate object
2. Create a menu-driven interface that will allow the user to select from a number of different reports of the potential years of life lost as described below:
    a. Total potential years of life lost due to all causes for all records in the report
    b. Total potential years of life lost by province (prompt user for province)
    c. Total years of life lost by sex (prompt user for male, female, or both sexes)
    d. Total years of life lost by year/date (prompt user for date)
    e. Total potential years of life lost by cause (prompt user for cause)
    f. Add a record to the database (save new record in the file.)
    g. Total potential years of life lost by cause for all records (Do last)
        i. Your program should be general enough that it will still work if with a new data file in the same format with new year values.

Hand In

A commented, object oriented solution to the programming task

Name Your Files as Follows:

- YourName-Unit4A1.py (Your main program)
- Record.py (The Record Class)

Evaluation Rubric

(Knowledge) Programming Concepts – Overall Knowledge and Understanding of the Required Programming Skills

12	10	9	7	6	4	0
Outstanding	Excellent	Good	Satisfactory	Adequate	Minimum Expectations Not Met	Not Demonstrated

(Thinking) Problem Solving – Overall Ability to Think Critically while Solving Computational Problems

4	3	2	1	0
Excellent	Good	Adequate	Minimum Expectations Not Met	Not Demonstrated

(Communication) Code Style – Overall Quality of the Internal Documentation and Naming Conventions

4	3	2	1	0
Excellent	Good	Adequate	Minimum Expectations Not Met	Not Demonstrated

(Application) Solutions – Overall Quality of the Solution to the Computational Problem

16	14	13	11	8	6	0
Outstanding	Excellent	Good	Satisfactory	Adequate	Minimum Expectations Not Met	Not Demonstrated