Individual Programming Assignment #3
CS 162: Introduction to Computer Science

Submit your assignment to Canvas

The purpose of the third program is to continue refining our use of functions and arguments in C++ and practice using **arrays of characters, structures, and external data files**. Again, our goal is to create programs with a small functions where main delegates to a series of functions where the real work takes place. In this programming assignment, you are **not** allowed to use global variables. Avoid using break *(unless working with a switch statement).* Limit your functions to no more than 30 statements of code (for executable statements… *not counting variable definitions, blank lines, lines with just curly brackets, or comments*). Never use a "return" in the middle of a loop!

Program Assignment:
As I sit here at my grid of computers, I think that my animals are the "stars" of my course videos! I have videos with one dog snoring and another who barks at just the wrong moment. Whenever I start a zoom meeting, the UPS truck will pass by and all of the dogs erupt into a chorus of song. In fact, they are singing right now as I write this assignment! Not only are pets great company, but they also can be very helpful – there are those that can sense seizure activity – and much more! It might be fun to keep track of what services are beloved animals can perform.

Your job for program #3 is to build a program maintain a list of animals that can perform service activities. We will use structures to us keep the data organized:

1. Breed (e.g., Schipperke)
2. Species (e.g., Canine)
3. Service (e.g., Sense Seizures)
4. Special information (any other information that would be helpful in deciding if this is the right type of animal for you
5. Pick one other field that is of your own choice

Using an array of structures, support up to a maximum of 10 animals in memory.

A large piece of this assignment is to use external data files, so that all animal information won't be lost next time the program is run. *Create a file called animal.txt. Make sure when you write your data to the file that there are delimiters ('|') between each field AND a newline at the end of an animal's information.*

Remember from the videos that with external data files, the information that you store in the files must be written in such a way that it is easy to read it back in. Also, make sure to keep all files in your "current working directory" on linux as the grader will not be able to replicate your directory structure.

IMPORTANT: Your program should support the following operations from main from a text-based menu; let the user continue to do this using a loop until they want to quit:

Task 1. Allow a new animal to be added to the array (up to 10)
Task 2. Allow the user to display everything
Task 3. Save the current list of animals to the external file
Task 4. Load animals from the file previously stored,
Task 5. Provide a menu interface to allow the user to select from (a) adding a new animal, (b) displaying all, (c) loading from a file, or (d) quit.
Task 6. Display all animals that match a particular species and breed.

******You are always welcome to do more! Really focus on making general purpose functions that can be re-used.***

Things you should know...as part of your program:

1. Make sure to prompt the user for any input requested. Make sure it is clear from your prompts what the user is expected to do.
2. You may **not** use any global variables in this program!
3. You may **not** use the string class – instead use arrays of characters. You **are allowed** to use the cstring library. Suggest using **strcmp**.
4. Make sure to use C++'s I/O (iostream library) for your input and output.
5. After each input operation, make sure to use **ignore** to remove the delimiters! This applies to the user input (istream) and file input (ifstream)
6. With external data files, first read before checking for end of file:
    i. Read
    ii. While (!infile.eof())
        1. Process what was read (store the data into the array of structs)
        2. Read again

CS162 - Checklist for First Week of Cycle

Cycle	Monday	Tuesday	Thursday	By Friday
First week Nov 1-5	DRAFT Algorithm Due by 7pm	Discussion: Share ALGORITHM to your Virtual Group	First Progress Submission: Tasks #1 and 2	Discussion Response
Second week Nov 8-12	Second Progress Submission Tasks #3-5	Discussion (Critique)	Submit Finished Assignment	Discussion Response

1. **Monday – Submit a typed DRAFT Algorithm**
    a. Due **by 7pm**
    b. Submit to **Assignments** on Canvas
    c. The Algorithm should be written in paragraph, using full English sentences (not code and not pseudo code)
    d. Write a paragraph about each function you plan to write
    e. We understand it may not be in its complete form yet!
    f. It should be at least 600 words

2. **Tuesday – Share YOUR ALGORITHM** with your Virtual Group
    a. Due **by 7pm**
    b. Submit it as a **Discussion Post** on Canvas

3. **Thursday – Submit First Progress Submission** as a .cpp file
    a. The progress submission must compile and have comments with your name and the purpose of the program
    b. Submit to Assignments on Canvas
        i. Learn to transfer the .cpp file from linux
    c. The progress submission must have these components:
        i. Functioning main program
        ii. Implement **Task #1 and #2**
        iii. Creating functions for these tasks is important
        iv. Code submitted should compile and run

4. **By Friday (earlier is better) Respond** to a flowchart posting
    a. Submit it as a response to a **Discussion Post** on Canvas
    b. Comment on at least one of your Virtual Group's
        i. What were some ideas that you found useful
        ii. Is there something missing that might be

important

CS162 - Checklist for First Week of Cycle

Cycle	Monday	Tuesday	Thursday	By Friday
First week Nov 1-5	Algorithm Due by 7pm	Discussion: Share Flowchart to your Virtual Group	First Progress Submission: Tasks #1 and 2	Discussion Response
Second week Nov 8-12	Second Progress Submission Tasks #3-5	Discussion (Critique)	Submit Finished Assignment Tasks #1-6	Discussion Response

5. **Monday - Submit** a **progress submission** as a .cpp file
   - **a.** Due **by 7pm**
   - b. Submit to Assignments on Canvas
     - i. Use an SSH program to transfer your program from linux
   - c. The progress submission must compile and run
   - d. Provide a **header comment** with a paragraph describing the purpose of the program
   - e. The progress submission must have these components:
     - i. Functioning main program
     - ii. Implement and Demonstrate **Tasks #1-5**
     - iii. The code should compile and run

6. **Tuesday – Critique the Plan** with your Virtual Group
   - a. Due **by 7pm**
   - b. Did the algorithm need to change?
   - c. Were there things you would do differently next time?
   - d. Ask a question of your Virtual Group

7. **Thursday - Submit** a **completed program** as a .cpp file
   - a. Due by **7pm**
   - b. **Remember comments and style are 20% of the grade**
   - c. Submit to Assignments on Canvas
     - i. Use an SSH program to transfer your program from linux
     - ii. Do not submit the a.out file.

8. **By Friday (earlier is better) Respond** to your Virtual Group

a. Submit it as a response to a **Discussion Post** on Canvas