

CPSC 122 Computer Science II

[Gonzaga University](#)

[Daniel Olivares](#)

PA5 – Pet Store 3 Linked Lists (100 points)

Individual, non-collaborative assignment

Learner Objectives

At the conclusion of this programming assignment, participants should be able to:

- Implement and use a linked list ADT from a class definition.
- Implement and use linked list class member functions
- Use and manipulate pointers and pointer operators for a linked list class.
- Use Dynamic Memory Management in C++ to create a linked list.
- Use public and private class member variables and functions for a linked list class.

Prerequisites

Before starting this programming assignment, participants should be familiar with C++ topics covered in CPSC 121 including (but not limited to):

- Conditional statements
- Implement loops to repeat a sequence of C++ statements
- Define/call functions
- Perform file I/O
- Work with 1D and 2D arrays
- Include and use the String and Vector libraries
- Demonstrate the software development method and top-down design

Overview and Requirements

This program will build upon the previous programming assignments to read information about the “inventory” of a pet store from an input file, `petstoredata.csv`, and output a summary report to an output file, `petreport.txt`. This program will require reading in and storing the entire contents of the pet store inventory into a linked list abstract data type object, require use of dynamic memory management, processing the data stored in that linked list object, and then generating an output text report to both the console and an output file. Your program will be required to make use of private and public member variables and functions to process the pet data and compute portions of the summary report.

Input File Format

The input file remains the same format as PA1 and PA2. The input file is in a CSV (comma-separated values) format. You are **not** allowed to modify the input file from the CSV format. The first line of the file contains the four (4) column header labels: Pet Store, Pet Name, Pet Type, Days at Store. You can assume that the input

Note: when graded, your program will be tested with a different pet store input file that will follow the same format but will NOT have the same number of rows or pet stores as the example input file.

See the below class definition and use it as your starting point for completion of this assignment's main tasks. See the starter cpp file attached to this document (copy-pasting from this file may result in unexpected errors). Also make sure to see the following tips to help you use this class object.

[illegible]

```

/* insertAtEnd
    * accepts a pointer to PetStoreData variable, inserts the PetStoreData "node" at
    the end of the list */
    void insertAtEnd(PetStoreData* newStoredData);

/* addPetData
    * adds pet name, pet type, and number of days that pet has been at the store
    * to the vector of pet data for the matching store name. */
    void addPetData(string storeName, string pName, string pType, int numDays);

/* displayPetList
    * prints the contents of the PetStoreList in forward order * see example output */
    void displayPetList() const;

/* writePetList
    * writes the contents of the PetStoreList in forward order to an output file
    * see example output */
    void writePetList(ofstream& outfile);

/* storeInList
    * accepts a string store name, returns true if the store name is in this object's
    list of stores */
    bool storeInList(string name);

/* calculatePetSummary
    * calculates/updates this object's PetSummary values, * see PetSummary programmer-
    defined data type */
    void calculatePetSummary();

/* displayPetSummary
    * prints this object's PetSummary values of the PetStoreList, see example output*/
    void displayPetSummary() const;

/* writePetSummary
    * writes this object's PetSummary values of the PetStoreList to an output file
    * see example output */
    void writePetSummary(ofstream& outfile);

//3 BONUS class member functions - i.e. NOT REQUIRED
/* insertAtFront (BONUS)
    * accepts a pointer to PetStoreData variable, inserts the PetStoreData "node" at
    the front of the list */
    void insertAtFront(PetStoreData* newStoredData);

/* insertAtPosition (BONUS)
    * accepts a pointer to PetStoreData variable and a position in the list, zero-
    based index just like arrays/vectors, inserts the PetStoreData "node" at the
    position location in the list, returns false if the position was not a valid
    position (e.g., out of range) */
    bool insertAtPosition(PetStoreData* newStoredData, int position);

/* deleteStore (BONUS)
    * accepts the name of a store to remove from the list, deletes a store and all of
    its petData, returns false if the store was not found or if the deletion was
    unsuccessful */
    bool deleteStore(string nameOfStoreToRemove);
};

```

Tips for using this class:

1. You only need a single `PetStoreList` variable declared in your main file.
2. Adding pet store and pet data to your list involves 3 steps
 - a. Create a new node
 - b. Insert that node into your linked list
 - c. Add pet data to the appropriate node (i.e., the node containing data for that pet store) in your linked list.
3. `createNode()` requires additional use of the scope resolution operator in your member function definition due to the private access scope of the `PetStoreData` data type.

Required Variables

There are no required variables outside of the class definition member variables. Below are suggested variables you may need at some point in your code. You are free to use them or not.

Reminder: you are not allowed to use non-constant global variables for your solution. All modifiable variables must be within scope of `main()` or your programmer-defined functions.

SUGGESTED (here are a few suggested variables you may want to make use of)

```
//input file variables
ifstream inFile;
ofstream outFile;
string inputFileName = "petstoredata.csv";
string outputFileName = "petreport.txt";
vector<string> header; //vectors to store header row data
stringstream lineToParse; //Hint: you can make use of stringstream to parse
                           strings based on a token - see the in-class example
string line = ""; //store each line
//temp variables for parsing input file
string tempString = "", petType = "", petName = "", storeName = "";
int dayAtStore = 0;
double averageDaysAtStore = 0;
bool firstRow = true;
PetStoreList storeList; //The linked list object
```

Required Functions

There are no required functions outside of the class definition class member functions. *You are still required to create additional functions in order to demonstrate proper top-down design* (see **Gaddis Chapter 1.6 The Programming Process** and the **Software Development Method** sheet attached to this assignment) demonstrating modular programming with the use of functions (see **Gaddis Chapter 6 Functions**).

Specifications

The input file pet information in the file consists of:

1. Store name (*string*: "Fur Get Me Not" is the name of the first pet in the example file below)
2. Pet name (*string*: "Chris P. Bacon" is the name of the first pet in the example file below)
3. Animal type (*string*: "pig" is the animal type of the first pet in the example file below)
4. Days in store location (*integer*: 1 is the number of days the first pet has been at this store in the example file below).
 - o Days in store is guaranteed to be at least 1.

The **output** of your program has two parts:

1. Status messages and summary displayed **to the console**. Status messages simply let the user know what the program is currently computing (e.g., Processed a pig, "Chris P. Bacon" ... 1

day(s) on site at store "Fur Get Me Not"). Message summary summarizes the store and pet data. *See the example console output below for the required format and output messages.*

2. Pet store summary information **written to an output file**. Name your output file `petreport.txt`. *See the example output file below for the required format and data.*

Main tasks:

1. Read all pet inventory information from `petstoredata.csv` for an **unknown number of pet stores and pets** and output the processing information as it is read in from the input file.
 - a. *Note that you cannot assume the input file contains a fixed number of records pets!*
 - b. *You will need to store each store name, pet name, pet type, and days at the store in your single linked list object.*
 - c. *Hint: read the input file line by line and then parse each line for the column data. I recommend using **getline** and **stringstream** to solve this problem (see in-class examples).*
2. Keep track of the following information:
 - a. Total number of pets in the input file, e.g., in the example data we have 7.
 - b. Pet average days on site across all stores, e.g., *in the example data this should calculate to 31.*
Note that we are not reporting fractional days.
 - c. Shortest number of days a pet has been at the store (across all stores). *1 in the example source.*
 - d. Longest number of days a pet has been at the store (across all stores). *101 in the example source.*
3. Display your summary report to the console immediately after your "processing" output.
4. Write the summary report to your output file.

Tip: I encourage you to use the software development method to plan out your algorithm to solve this problem and **incrementally code-compile-test each portion of your program** as you implement it! It may take a little longer at the start but I promise you that it will save you time and headache in the long run!

Note: You should not need to use the string function `stoi()`, `stod()` or any variants for your solution. I am not going to forbid it but you need to be able to understand and apply techniques used to overcome the issue encountered when switching between the extraction operator (`>>`) and the string `getline()` function.

(Bonus 10pts)

You must state in your top level comment block that you are attempting the extra credit. Implement the bonus 3 class member functions: `insertAtFront()`, `insertAtPosition()`, and `deleteStore()` for 1 point, 3 points, and 3 points. For the final 3 points, **AFTER the last line in the console output (and after writing your summary to the output file)**, output to the console:

1. **"*****BONUS TASK*****"** to clearly indicate this part of the console output is BONUS
2. Manually create 2 new nodes containing a new pet store name and 2 new pets and their data for each bonus insert function (so 2 new pet stores and 4 new pets total).
3. Insert the first new pet store data at the front, insert the second new pet store data as the new third item (index 2) in the linked list.
4. Update your pet summary stats again
5. Display both the summary and list to the console.

Submitting Assignments

1. Submit your assignment to the Canvas course site. You will be able to upload your three source files (**two .cpp files and one .h file**) to the PA assignment found under the Assignments section on Canvas. **You are REQUIRED to use the three-file format** for this submission. Use the "+ Add Another File" link to allow choosing of three files (see reference image below).

2. **Your project must compile/build properly.** The most credit an assignment can receive if it does not build properly is 65% of the total points. Make sure you include all necessary libraries, e.g. string, ctime, etc. (though this does not mean add every single library you've ever used!)
3. Your submission should only contain your three source files (plus any bonus submission files). You may submit your solution as many times as you like. The most recent submission is the one that we will grade. *Remember to resubmit all three files if you submit more than once.*
4. Submit your assignment early and often! You are NOT penalized for multiple submissions! We will grade your latest submission unless you request a specific version grade (request must be made prior to grading).
5. If you are struggling with the three-file format, I recommend you complete the program in one file and submit that (working) version first, then convert it to the three-file format. **A working one-file format program is worth more points than a broken three-file format program!**

Note: *By submitting your code to be graded, you are stating that your submission does not violate the CPSC 122 Academic Integrity Policy outlined in the syllabus.*

Grading Guidelines

This assignment is worth 100 points. Your assignment will be evaluated based on a successful compilation and adherence to the program requirements. We will grade according to the following criteria:

- 5 points for properly reading in csv data
- 5 points for correctly calculating main tasks stats
- 5 points for correct console output and correct file output
- 15 points for PetStoreList() and ~PetStoreList() implementation
- 5 points for createNode() implementation
- 10 points for insertAtEnd() implementation
- 10 points for addPetData() implementation
- 7 points for displayPetList() and displayPetSummary() implementation
- 7 points for writePetList() and writePetSummary() implementation
- 16 points for storeInList() and calculatePetSummary() implementation
- 5 points for using the three-file format and guard code
- 10 pts for adherence to proper programming style and comments established for the class and for demonstration of top-down design and modular programming using functions

Example input file (petstoredata.csv)

```
Pet Store,Pet Name,Pet Type,Days at Store
Fur Get Me Not,Chris P. Bacon,pig,1
Pick of the Litter,Mary Puppins,dog,17
For Pet's Sake,Jean-Clawed Van Damme,cat,60
Fur Get Me Not,Jack Meower,cat,24
Fur Get Me Not,Severus Snake,snake,12
For Pet's Sake,Lame Duck,duck,5
Pick of the Litter,Barktholamew,dog,101
```

Expected Console Output

Processed 4 header columns: Pet Store, Pet Name, Pet Type, Days at Store

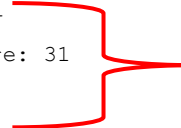
Processed a pig, "Chris P. Bacon" ... 1 day(s) on site at store "Fur Get Me Not"
Processed a dog, "Mary Puppins" ... 17 day(s) on site at store "Pick of the Litter"
Processed a cat, "Jean-Clawed Van Damme" ... 60 day(s) on site at store "For Pet's Sake"
Processed a cat, "Jack Meower" ... 24 day(s) on site at store "Fur Get Me Not"
Processed a snake, "Severus Snake" ... 12 day(s) on site at store "Fur Get Me Not"
Processed a duck, "Lame Duck" ... 5 day(s) on site at store "For Pet's Sake"
Processed a dog, "Barktholamew" ... 101 day(s) on site at store "Pick of the Litter"

All pet store data processed!

Generating summary report...

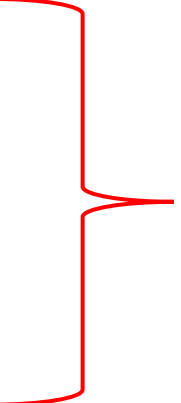
Done!

```
-----  
Total number of pets: 7  
Average number of days at pet store: 31  
Shortest stay at pet store: 1  
Longest stay at pet store: 101
```



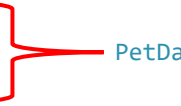
PetSummary

```
-----  
Pet Store Name: Fur Get Me Not  
*****  
| Number of pets: 3  
| Pet Information:  
> Pet 1 Name: Chris P. Bacon  
> Pet Type: pig  
> Number of days at store: 1  
-----  
> Pet 2 Name: Jack Meower  
> Pet Type: cat  
> Number of days at store: 24  
-----  
> Pet 3 Name: Severus Snake  
> Pet Type: snake  
> Number of days at store: 12  
-----
```



PetStoreData

```
-----  
Pet Store Name: Pick of the Litter  
*****  
| Number of pets: 2  
| Pet Information:  
> Pet 1 Name: Mary Puppins  
> Pet Type: dog  
> Number of days at store: 17  
-----  
> Pet 2 Name: Barktholamew  
> Pet Type: dog  
> Number of days at store: 101  
-----
```



PetData

```
-----  
Pet Store Name: For Pet's Sake  
*****  
| Number of pets: 2  
| Pet Information:  
> Pet 1 Name: Jean-Clawed Van Damme  
> Pet Type: cat  
> Number of days at store: 60  
-----  
> Pet 2 Name: Lame Duck  
> Pet Type: duck  
> Number of days at store: 5  
-----
```

Expected File Output (petreport.txt)

Note: The output file contains EXACTLY the same information as the console output minus the “processing” portions of the output and starts with a different title text seen below:

Pet Store CSV Summary Report

Total number of pets: 7
Average number of days at pet store: 31
Shortest stay at pet store: 1
Longest stay at pet store: 101

Pet Store Name: Fur Get Me Not

| Number of pets: 3
| Pet Information:
> Pet 1 Name: Chris P. Bacon
> Pet Type: pig
> Number of days at store: 1

> Pet 2 Name: Jack Meower
> Pet Type: cat
> Number of days at store: 24

> Pet 3 Name: Severus Snake
> Pet Type: snake
> Number of days at store: 12

Pet Store Name: Pick of the Litter

| Number of pets: 2
| Pet Information:
> Pet 1 Name: Mary Puppies
> Pet Type: dog
> Number of days at store: 17

> Pet 2 Name: Barktholamew
> Pet Type: dog
> Number of days at store: 101

Pet Store Name: For Pet's Sake

| Number of pets: 2
| Pet Information:
> Pet 1 Name: Jean-Clawed Van Damme
> Pet Type: cat
> Number of days at store: 60

> Pet 2 Name: Lame Duck
> Pet Type: duck
> Number of days at store: 5
