CPSC 2430 Data Structures

Winter Quarter 2023

Assignment 1 **Due: 10:00pm, Thursday, Jan 11, 2024**

Assignment #1 is to practice basic data structures you have learned before. In this assignment, we focus on IO, variable allocation, functional decomposition, and error handling.

Animal Shelter Promotion System

Suppose we want to build an animal shelter, which holds only dogs and cats, and operates on a strictly "first in, first out" basis. Adopters must follow this policy to adopt an animal:

1. If an adopter has no preference, this adopter must adopt the "oldest" (based on the arrival time) of all animals at the shelter.
2. If an adopter wants to adopt a dog, this adopter must adopt the "oldest" dog at the shelter (based on the arrival time).
3. If an adopter wants to adopt a cat, this adopter must adopt the "oldest" cat at the shelter (based on the arrival time).
4. An adopter cannot select which specific animal they would like to take.

Create an animal shelter management system to implement this process.

The system should include the following functions:

1. Add an animal to the system – when adding an animal, the system needs to record animal's category (dog or cat) and animal's name (you can name the animal whatever you like).
2. Get an animal from the system. The adoption policy is defined as above. When retrieving an animal from the system, the system needs to tell the adopter the category and name.

Besides, your system should also have the following features:

1. User interface. When testing your system, a user interface should be provided to allow users to add/get an animal and quit the system.

2. Error handling. The systems should include error handling. For example, when an adopter wants to adopt a dog/cat while 0 dog/cat is in the shelter, the system should give a message (such as “There is no dog/cat in the shelter now”) to the user. When an adopter does not specify the category, but no animal is in the shelter, the system should give a message too (such as “There is no animal in the shelter now”).

An example of a user interface and its error handling (you may choose to have different, choices, strings, etc, up to you as long as all the expected functionality is there):

```
Welcome to the animal shelter. What would you like to do?
1. Add a CAT to the shelter
2. Add a DOG to the shelter
3. Adopt a CAT from the shelter
4. Adopt a DOG from the shelter
5. Adopt any animal from the shelter
6. Exit
1
Please enter the CAT's name:
Link
Welcome to the animal shelter. What would you like to do?
1. Add a CAT to the shelter
2. Add a DOG to the shelter
3. Adopt a CAT from the shelter
4. Adopt a DOG from the shelter
5. Adopt any animal from the shelter
6. Exit
3
Congrats on adopting a cat, their name is: Ozzy
Welcome to the animal shelter. What would you like to do?
1. Add a CAT to the shelter
2. Add a DOG to the shelter
3. Adopt a CAT from the shelter
4. Adopt a DOG from the shelter
5. Adopt any animal from the shelter
6. Exit
5
Congrats on adopting a dog, their name is: Gus
Welcome to the animal shelter. What would you like to do?
1. Add a CAT to the shelter
2. Add a DOG to the shelter
3. Adopt a CAT from the shelter
4. Adopt a DOG from the shelter
5. Adopt any animal from the shelter
6. Exit
4
Sorry, no dogs are available for adoption today
```

In this project, you are required to use one or more of the following structures: linked-list, stack, or queue. You may use the C++ Standard Template Library (STL). More information on STL container classes: https://www.cplusplus.com/reference/stl/

Project 1 Submission

Name your source code file **assignment1.cpp**

Your assignment1.cpp should contain all necessary class and function definitions, as well as main function to drive the user interface and tests. Before submission, you should ensure your program compiles without errors when running

```
g++ -Wall -Werror -pedantic -std=c++11 assignment1.cpp
```

on the cs1 server, and it should be tested extensively. Your assignment receives zero if your code cannot be compiled and executed.

You can submit your program multiple times before the deadline. The last submission will be used for grading. To submit your assignment, run the script below from the directory containing your assignment1.cpp file (assuming your files are on cs1.seattleu.edu):

```
/home/fac/mjilani/submit/24wq2430/assignment1_submit
```

Assignment 1 Grading Breakdown

Breakdown	Points	Note
Add an animal	10	Add an animal to the system successfully **based on the arrival time**.
Get an animal	45	Get the oldest animal if the adopter has no preference: 15 pts Get the oldest dog (**based on arrival time**) if an adopter chooses to adopt a dog: 15 pts Get the oldest cat (**based on arrival time**) if an adopter chooses to adopt a cat: 15 pts
User Interface	15	Provide a user interface to allow users to 1. add an animal: 5pts 2. get an animal: 5pts 3. quit the system: 5 pts
Error Handling	15	Give users messages when there is no dog to adopt when adopters choose a dog to adopt: 5pts Give users messages when there is no cat to adopt when adopters choose a cat to adopt: 5pts Give users messages when there is no animal to adopt: 5pts
Coding	15	Clean, well-commented code. No unsolicited output messages, such as testing & debugging messages. No global variables. Free up any dynamically allocated memory.