**Program 5 – SMU Animal Daycare**

Due Monday April 22 at 11:59pm

The Task

Your task is to create a pet management software solution for your new Pet Daycare Business! Your daycare center will house a variety of different animals like Cats, Dogs and Rabbits. You will create a program that reads in a list of pets that are currently in daycare and have an interactive mode where a user can choose to drop off a pet or pick up their pet. You will use a Linked List to store the list of Pets and utilize Inheritance and Polymorphism in order to store and represent pets in your system.

**Part I – Reading in the input file.**

The first step to this program is to read in the list of animals that are currently in the system already. You must use **command line arguments** to provide the file name of the input file you are reading in. Here is an example input file:

*Input.txt*

cat,Elroy,1,false

dog,Charlie,2,bone

rabbit,Squeekers,3,5

Each line contains information about one animal. We will discuss this information below:

**Part II - The Animals**

As part of this program, you are required to use inheritance and polymorphism. To do so you will implement an *Animal* base class that contains information that pertains to all animals. In your shelter ALL animals have a *name,* an *age,* and a *dailyFee* that represents the fee for keeping the animal at daycare. The fee for each animal is as follows:

Cat: $25

Dog: $50

Rabbit: $15

Your derived classes will be the specific types of animals housed in your daycare: *Cat*, *Dog* and *Rabbit*. Each animal has additional data that only corresponds to that specific animal.

A Cat has a property called *isOutdoor*. This value will be *true* if the cat is an outdoor cat and *false* if the cat is an indoor cat.

The input line for a cat is: *cat,<name>,<age>,<isOutdoor>*

A Dog has a property called *favoriteToy.* This value is a string that will hold the name of his favorite toy (Ex: A bone, A ball, etc)

The input line for a dog is: *dog,<name>,<age>,<favoriteToy>*

A Rabbit has a property called *numberOfCarrots*. This value is an integer that represents the number of carrots a rabbit is allowed to be fed per day.

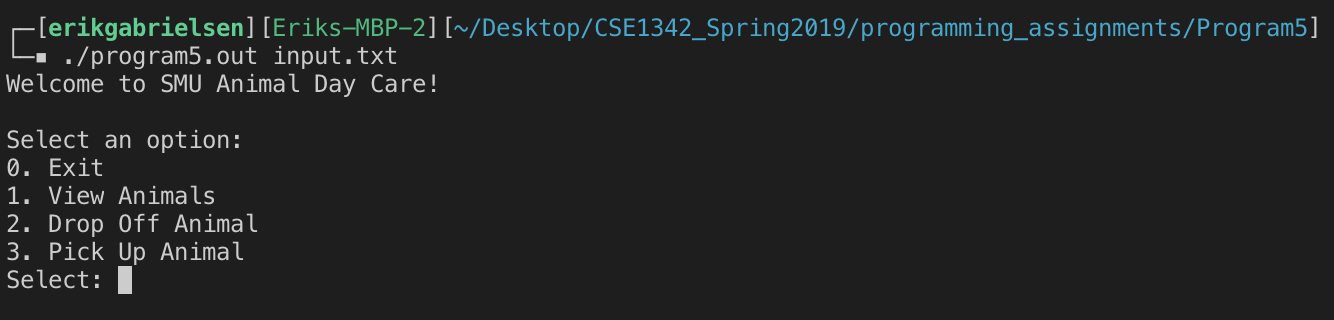
The input line for a rabbit is: *rabbit,<name>,<age>,<numberOfCarrots>*

***HINT***: Use a UML Diagram to outline your class infrastructure given the above information!

**Part III – Pet Daycare User Interface**

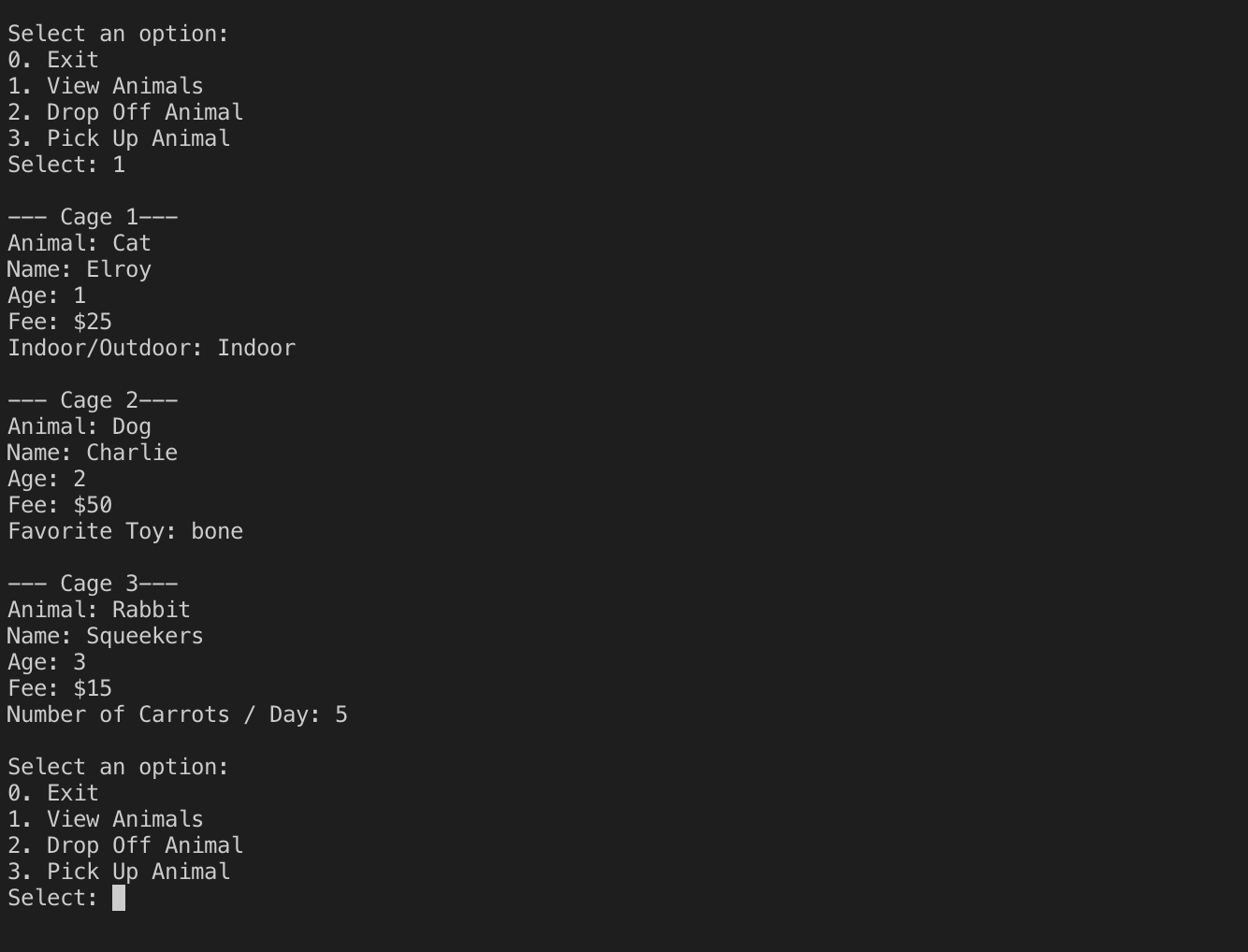
After reading in the input file, you will prompt the use to select from a list of options:

*Sample output:*

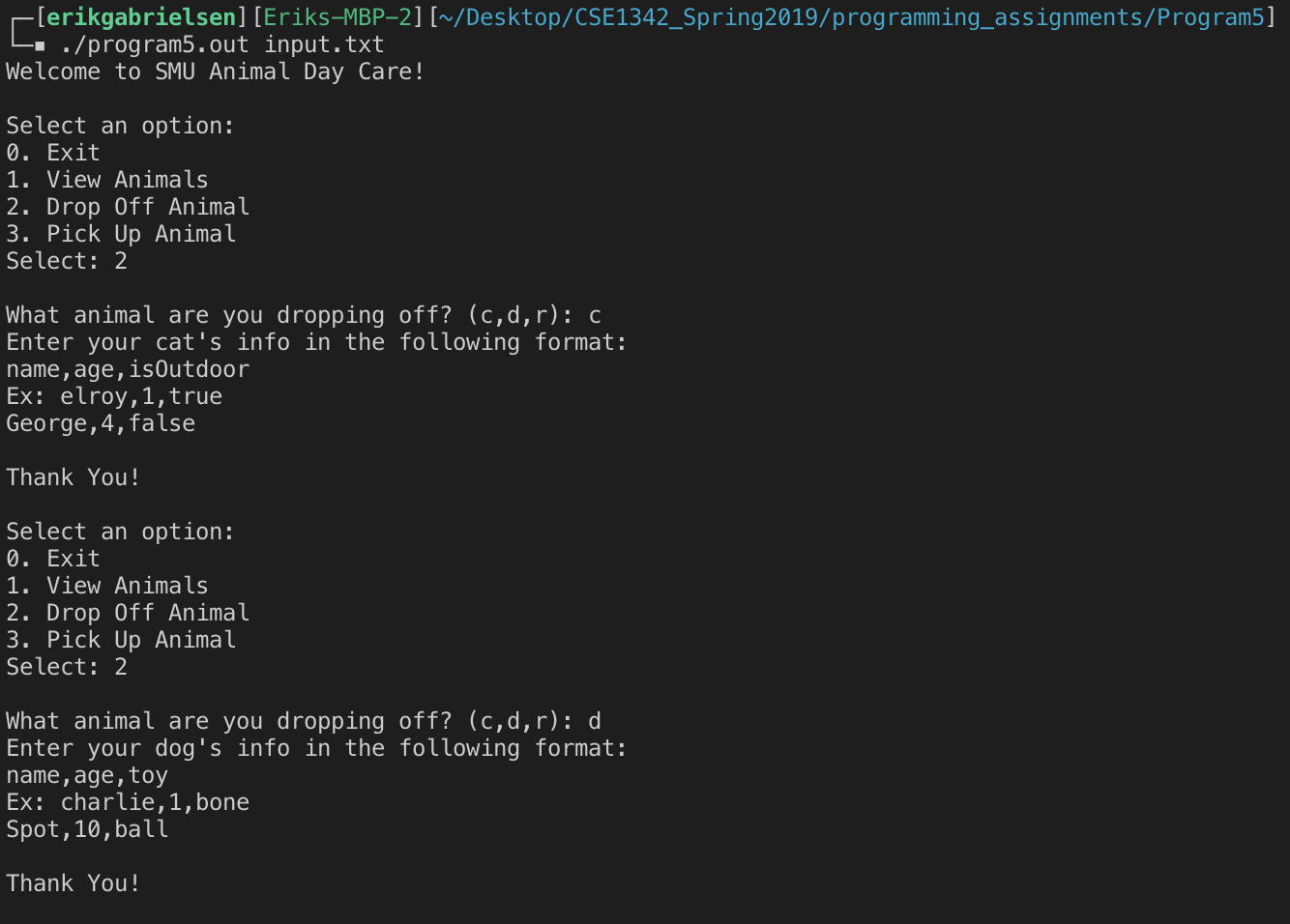
**

**Option ‘0’ –** when the user enters 0 (zero) the program will end.

**Option ‘1’ –** when the user enters 1, the user will be given a list of all animals currently at the shelter. Here is the example output for option 1:

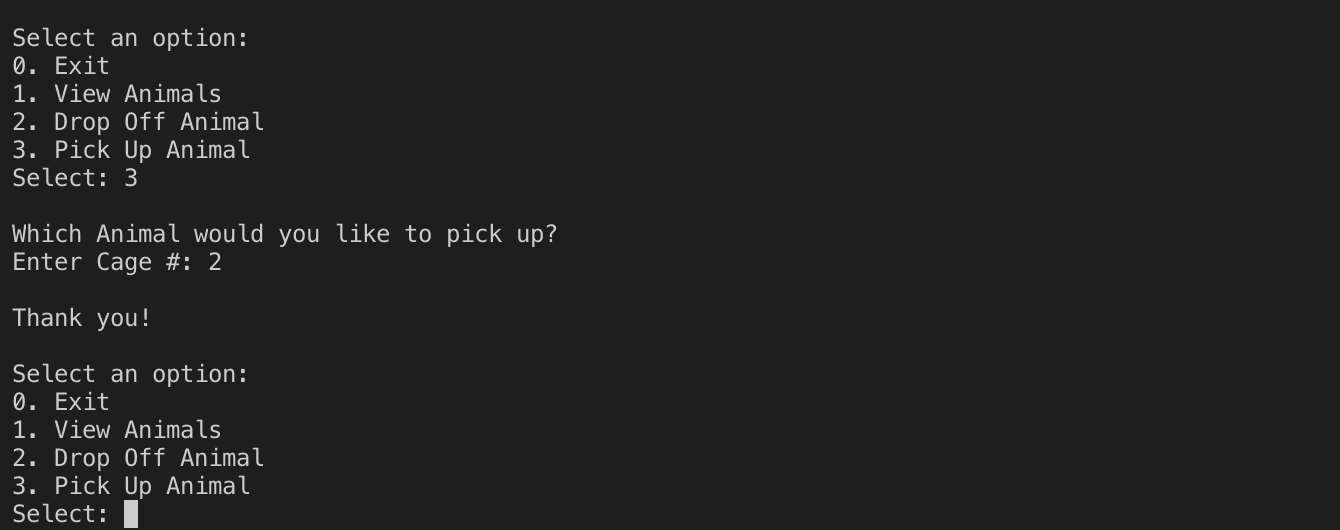


**Option ‘2’ –** When the user enters 2, the user will be prompted for what type of animal they are dropping off (c/d/r) and then provide the information for that animal. You may assume that a user will enter in the correct information in the correct format for each animal type (cat, dog, rabbit). Here is an example output for Option 2:



Once the user submits the information you will create a new animal of that type and add it to the linked list that contains all the animals in the shelter. If the user selects option 1 again, they should be provided with an updated list of all animals plus the ones the user has just entered appended to the end of the list.

**Option ‘3’.-** When the user enters 3, the user will be prompted for the cage number of the animal that they are picking up. When the user enters the number that animal will be removed from the linked list.



Implementation

Your Program must consist of .h and .cpp files for each class that you define. For this program you should have an Animal class, a Cat class, a Dog class, and a Rabbit class. Your Cat, Dog, and Rabbit class will inherit data members and member functions from your Animal base class.

In your Animal class, you will need to define a virtual *print* method that will be overridden in your derived classes (Cat,Dog,Rabbit). Each one of these classes should provide their own *print* method that will be called when iterating through the linked list to print out all animals in the shelter.

All data members in your class definitions must be *private* or *protected.* No public data members are allowed! This means that you must provide getters and setters for your data members so that you may access them outside of your class definition and implementation.

In your program you will be storing all animals in your own custom Linked List. You may NOT use the list ADT provided by the C++ STL. For your linked list implementation you will at a minimum need an *insert* method, a *removeAt* method, and a *print* method. If implemented as a class you must also include a separate .h and .cpp file for your LinkedList class.

Grading

Your Program will be graded using the following for a total of 100pts:

Uses Inheritance and Polymorphism correctly…………..20

Linked List Implementation……………………………………….20

Program Functionality……………………………………………….40

Code Formating/Styling/FileStructure……………………….20

What to Submit

You will be turning in a .zip file called <LastName\_FirstInitial>Program5.zip to canvas containing all .h and .cpp class files, your input txt file, your main *program5.cpp* file, and your executable *program5.out.*

To compile: g++ -o program5.out \*.cpp

To run: ./program5.out input.txt