**Final Part 2**

Directions:

Published text will be allowed (open book policy) along with handouts posted on Titanium course webpage and copies of your personal program listings from earlier assignments. No electronic media (hard drives, flash drives, CD’s, etc. ) including cell phones will be allowed except for authorized software (compiler, word processor). Furthermore, no access to the Web will be allowed during the exams except for accessing the Titanium site.

You might find yourself under some time pressure in this examination. Please check the point value for each problem so that you do not spend more time on one problem than it is worth. Please make sure to read each problem carefully before working on it. You will also get partial credit.

**Academic Dishonesty Policy**

Academic dishonesty includes such things as cheating, plagiarism, and helping someone else commit an act of academic dishonesty. Cheating is defined as the act of obtaining or attempting to obtain credit for work by the use of any dishonest, deceptive, fraudulent, or unauthorized means. Any test, paper or report submitted by you and that bears your name is presumed to be your own original work. The consequences of cheating and academic dishonesty – including a formal discipline file, possible loss of future employment opportunities – are simply not worth it. Additional information on this policy is available from University Policy Statement 300.021.

Please PRINT your name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please SIGN your name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please make sure to read each problem carefully before working on it. **Every file you turn in must have your name at the beginning as documentation. NO OTHER DOCUMENTATION IS REQUIRED unless specifically stated.**

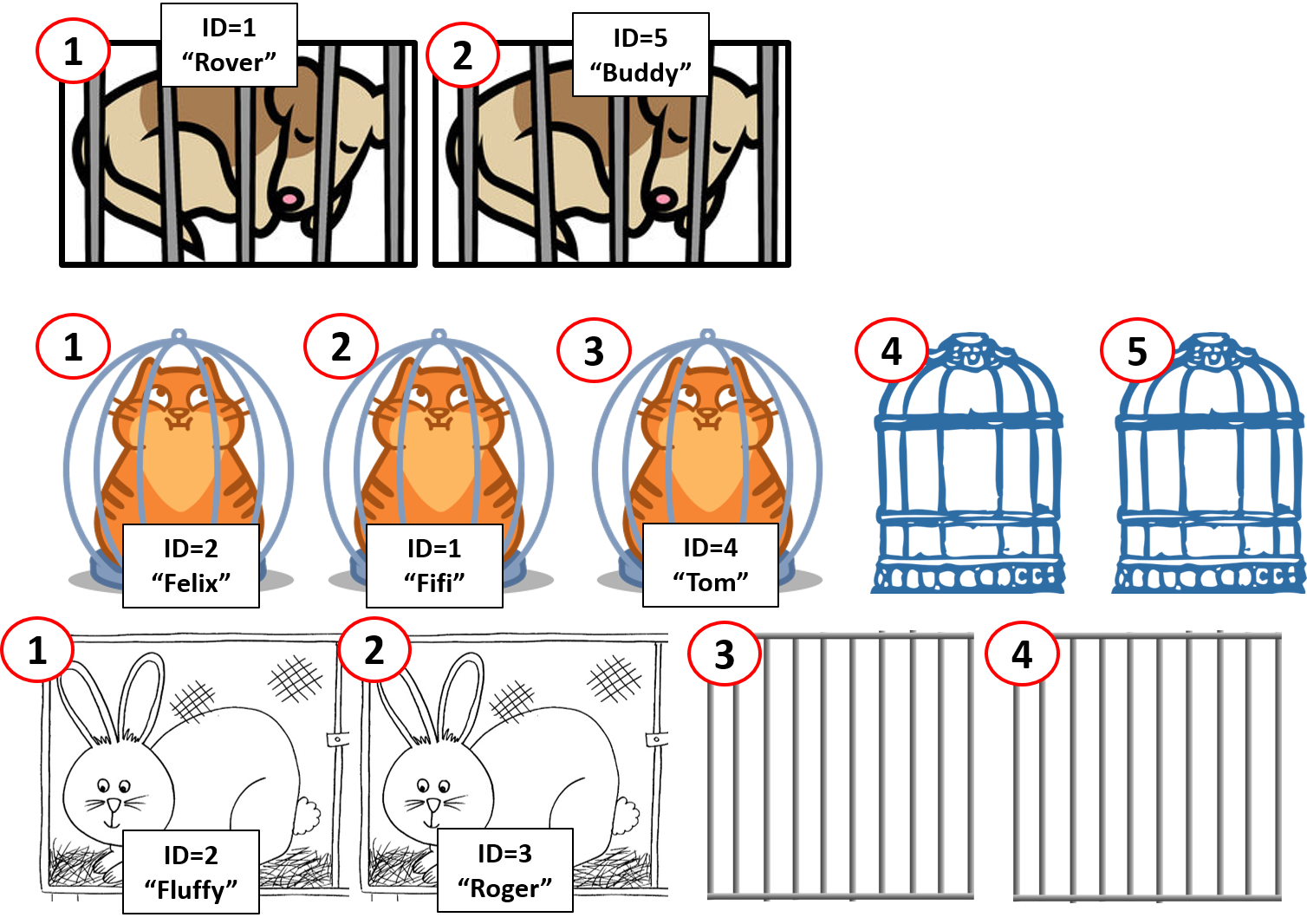
**Total points: 40**

**Extra credit: 5 more points**

**Write a complete C++ program as described below. Documentation is required; see section below.**

Write an interactive program for a small animal shelter. The shelter has a limited number of cages to keep dogs, cats, and rabbits. Each cage can contain only one animal. Separate cages are maintained for dogs, cats, and rabbits. The two main tasks for the program are:

1. Animal intake: A person brings in a dog, cat, or rabbit. If there is an empty cage available for the type of animal, then the animal’s information (id, name, and age) is recorded and number of remaining empty cages is updated. If an empty cage is not available, then the animal cannot be taken in.
2. Animal adoption: A person comes in to adopt a particular animal. The person provides the type of animal (dog, cat, or rabbit) and id of an animal. That animal information is removed from the system and number of empty cages is updated.

****

**Program Specifications:**

When the program first starts, it will first ask for the number of available cages for dogs, cats, and rabbits (i.e., three numbers). You may assume that the number of available cages will never be greater than 100.

The program will then print a menu and the user is asked to enter one of the following letters corresponding to the 4 choices:

**N:** this stands for iNtake.

The program will then ask the user to enter the type of animal: D for dog, C for cat, R for rabbit. If there are no empty cages for that type of animal, a message will inform the user and the program return to the main menu. If there is an available cage, the program will then ask the user to enter the animal ID (an integer), name, and age. (Note that the ID is unique only to animals of the same species, i.e., a dog and cat may have the same id but two dogs cannot have the same id.)

The program will then enter the information in the system by place it on the appropriate list.

**A:** this stands for Adoption.

The program will ask for the type of animal: D for dog, C for cat, R for rabbit and the animal ID. If such an animal cannot be found, a message will inform the user, and the program return to the main menu. If the animal is found, the animal information is removed from the appropriate list.

**P:** this stands for Print.

The program will print: First, the list of all dogs in cages (with their ID, name, and age); Second, the list of all cats; and Third, list of all rabbits.

**Q:** this stands for Quit.

The program quits. Unless “**Q**” is chosen, the main menu should be shown again.

**Resources**:

On the CS 301 Titanium site, if you go to the “Final Part 2” section, you will find all the files given for this problem. These include two files, Animal.h and Animal.cpp, that implement a simple animal record class with only ID, name, and age as data members.

You are also given files containing templates for a LinkedList class along with a test program that instantiates the LinkedList using a few Animal records. All function bodies are given. You may use none, any, or all of them if it will help you. You are also free to revise any of these files if you wish.

**Required documentation:**

Your name at the top for each file you write or revise.

A comment in the main cpp file that lists what, if any, data structure(s) you are using and for what purpose.

If any variable name does not make it immediately clear what its role is, that variable must have comments whenever it is used.

All functions, **including the main**, must have a brief documentation that states **what the function does**, and **the roles played by each of the parameters**, if the function has parameters.

Other documentation is optional. Any unusual or tricky code should be documented, but please do not just repeat the code.

You should name your source file **part2LastNameInitial.cpp** and submit it to the Titanium Final Part 2 assignment. Please also turn it in any additional files that you wrote or revised**,** adding **LastNameInitial** to the original file names.

**Extra credit: 5 more points**

Add code so that all information in the main system will be saved in a binary file called **Animals.bin** in the Quit option. Make sure that the information will be written in such a way that the next time the program starts, they can be put back into the system as before. Note that the program has to save all of this information: number of cages for each animal, and the list of animals currently in each of the three types of cages.

**Sample run:**

Number of dog cages: 3

Number of cat cages: 5

Number of rabbit cages: 4

===== Main Menu =====

N: iNtake

A: Adopt

P : Print

Q : Quit

your choice? N

Animals to intake:

D: Dog

C: Cat

R: Rabbit

What kind of animal? D

ID? 1

Name? Rover

Age? 5

===== Main Menu =====

N: iNtake

A: Adopt

P : Print

Q : Quit

your choice? N

Animals to intake:

D: Dog

C: Cat

R: Rabbit

What kind of animal? D

ID? 2

Name? Buddy

Age? 4

===== Main Menu =====

N: iNtake

A: Adopt

P : Print

Q : Quit

your choice? N

Animals to intake:

D: Dog

C: Cat

R: Rabbit

What kind of animal? D

ID? 3

Name? Yeller

Age? 10

===== Main Menu =====

N: iNtake

A: Adopt

P : Print

Q : Quit

your choice? N

Animals to intake:

D: Dog

C: Cat

R: Rabbit

What kind of animal? D

No more space for dogs

===== Main Menu =====

N: iNtake

A: Adopt

P : Print

Q : Quit

your choice? N

Animals to intake:

D: Dog

C: Cat

R: Rabbit

What kind of animal? C

ID? 1

Name? Fifi

Age? 5

===== Main Menu =====

N: iNtake

A: Adopt

P : Print

Q : Quit

your choice? P

List of dogs

ID: 1

Name: Rover

Age: 5

ID: 2

Name: Buddy

Age: 4

ID: 3

Name: Yeller

Age: 10

List of cats

ID: 1

Name: Fifi

Age: 5

List of rabbits

===== Main Menu =====

N: iNtake

A: Adopt

P : Print

Q : Quit

your choice? A

Animals to adopt:

D: Dog

C: Cat

R: Rabbit

What kind of animal? D

Give id of dog? 2

Adopting out:

ID: 2

Name: Buddy

Age: 4

===== Main Menu =====

N: iNtake

A: Adopt

P : Print

Q : Quit

your choice? P

List of dogs

ID: 1

Name: Rover

Age: 5

ID: 3

Name: Yeller

Age: 10

List of cats

ID: 1

Name: Fifi

Age: 5

List of rabbits

===== Main Menu =====

N: iNtake

A: Adopt

P : Print

Q : Quit

your choice? Q