ENSF 409
Winter Semester 2021
Assignment 04

Assignment Instructions

Complete and submit exercise 10.3 (Lesson 10). This assignment is due February 8, 2021 by 5:59 PM Mountain Time.

General Instructions

Academic Integrity and Collaboration

This is an individual assignment. Your submission must be your own original work. You may not work with others to complete the assignment, although you may collaborate on exercises which are not submitted. It is not academic misconduct to ask for help on the discussion boards, but you may not copy code or complete answers from your peers.

Submission

You must submit your assignment in the specified format.

Your assignment must consist of a single file, in PDF or PNG format. It is preferred that you submit a solution drawn on a computer, but a scan of a handwritten solution will also be accepted, provided it is legible.

Use your student ID as the name of the file. For example, if your student ID is *12345678* the file could be named *12345678.png* or *12345678.pdf*. The extensions may be written in capital letters or lower case.

You must submit your assignment to the appropriate D2L dropbox folder. You may submit multiple times before the assignment deadline, but only the last submission will be graded. Previous uploads are discarded by D2L.

Deadline

All homework assignments are due at 17:59 (5:59 PM) Mountain Time. You are responsible for the conversion between Mountain Time and your local time zone, if applicable. Be aware that the switch from Mountain Standard Time to Mountain Daylight Time occurs during the term.

It is recommended that you do not leave your submission until the last minute in case of technical issues. Once the dropbox folder closes, you will not be able to submit your assignment. Late submissions will not be accepted.

If there is a technical problem and you are unable to submit via the dropbox, you may email a link to download your .zip folder from OneDrive. The email must be received by Dr. Barcomb and Dr. Marasco before the assignment deadline and the OneDrive folder should not show any changes to the submission file after the deadline has passed.

ENSF 409

Exercises - Lesson 9

The following exercises are described in the video for Lesson 9.

Exercise 9.1

1. Modify the horse and rider code.
2. Create methods for accessing the skill and level data. Set and access the new skills format in main().
3. In the class Rider, change the registration number to be an alphanumeric String instead of a number. Was it easier than changing the code that accessed skills?
4. Ensure the registration number is always a capital letter followed by four digits. How many places did you need to change the code?

Tip: Remember the lesson on primitive wrappers when you evaluate whether or not a character is a letter.

Exercise 9.2

1. Determine which attributes you would include for an abstraction of “plant” from the perspective of a garden center.

Potential attributes:

- blossomColour
- bulkPrice
- clade
- commonName
- cultivar
- dateReceived
- family
- floweringMonth
- genus
- kingdom
- needleLength
- order
- originatingNursery
- pollenCount
- preferredAcidity
- preferredLight
- retailPrice
- species
- toxicity
- wateringFrequency

Tip: Think of several different types of plants and what they have in common.

Exercise 9.3

1. Create a "has a" hierarchy for a plant. Include at least six components, and at least two levels of hierarchy.
2. Create a "is a" hierarchy for a plant. Include at least six components, and at least two levels of hierarchy.

ENSF 409

Exercises - Lesson 10

The following exercises are described in the video for Lesson 10.

Exercise 10.1

1. Create a simplified UML class (only including relevant data members) diagram representing the specified relationships.
2. Specify cardinality and roles.

- A **Horse** must be associated with a **Farm**.
- A **Farm** may have livestock, namely **Horse**[s].
- A **Farm** must be associated with between 2 and 20 barn **Cat**[s].
- A barn **Cat** doesn't know it is associated with a **Farm**; as far as it is concerned, its territory extends in all directions.
- A **Farm** is always associated with some **Person** [People] who are workers.
- Each **Person** works at only one **Farm**.
- There is a management structure where each worker has just one (or none, in the case of the lead worker) manager, but each manager supervises up to 10 people.

Tip: Check your work against the diagram in the repository. You may have used different names for variables and roles.

Note: Class names are generally singular, because each instance is singular.

Exercise 10.2

1. Create a simplified UML class (only including relevant data members) diagram representing the specified relationships.
2. Specify cardinality and roles.

- A **House** has a **Kitchen**.
- A **House** has at least one **Bathroom**.
- A **Bathroom** might have, at most, one **Bathtub**.
- A **House** has a single **Roof**.
- A **House** could have **Tenant**[s].
- A **Tenant** only rents one **House**.
- The relationship between a **House** and a **Tenant** is mediated through a **RentalAgreement**.
- A **House** has **Door**[s].
- A **House** might have a **Deck**.

Exercise 10.3

1. Create a complete (with attributes and methods) UML diagram of the scenario.
2. Include cardinality.
3. Use the named association approach for describing any asymmetric reflexive associations.
4. Use the analysis phase for representing any association classes.
5. Methods and attributes are public unless otherwise stated.
6. Use terms which appear in the text when selecting names.

Tip: The completed diagram should contain nine classes.

Important: Ensure class names are singular.

The owners of YYC Pet Resort, a rapidly growing boarding facility for cats and dogs, need to design an application to organize their client and facility information.

According to the company owners:

"The pet is at the core of our business. We take special care to learn everything we can about our boarders.

"Of course, clients are important, too. We try to retain a relationship with them regardless of their situation. Even when a pet has passed away, we keep their name, phone number, and address on file. We want to be sure that if they adopt a new pet they will think of us for their boarding needs.

"We're very careful to keep a care profile for each pet, which includes feeding instructions, a medication list, and medication instructions. We make a print-out of the care instructions for all pets in our care, which are posted so that staff don't have to search for this information. We can also pull up the complete care profile on the computer."

Lalitha Chitnis, manager of the northwest Calgary store, described some of the functionality she needs to perform her duties.

"Whenever a new employee joins, I enter their name and then I have to assign them an ID number. Each employee has a primary store they're associated with, and we can also see that information in the store profile. When I look up the store, I see not only the store's name, address, and phone number, I also see the list of all employees assigned to the store."

Thomas Walls, head of advertising, specified additional requirements:

"The rewards profile is really important to the business. We keep the rewards number and points total. It's an important part of the client profile, but of course not every client is part of the rewards program, though we really encourage them to enroll. When they're not enrolled, I just see their reward number as 'Not enrolled'. They get 10 points just for enrolling.

"They get points for every time they board with us, and in the end that can translate to things like a complimentary grooming. Once a month, we pick a rewards profile at random and give a free day of boarding to the winner, regardless of their rewards points. This makes it worthwhile for all clients to enroll in the program, even if they don't board often enough to earn many points. When I'm looking at a client's information, I can immediately see their rewards profile and tell them how close they are to earning a reward."

Slobodan Klasinc, manager of the Kensington Calgary store, provided the following information:

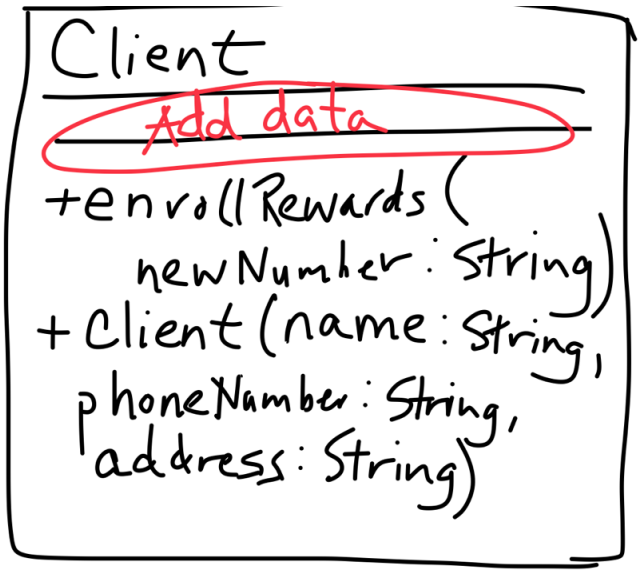
"Our store is the largest in Calgary and sets the standards for all the others. We have shift managers who are responsible for up to ten employees but our employees work the same schedule every day so they only have one manager. Shift managers are responsible for checking at the start of the shift the list of pets assigned to each employee. All employees are assigned several pets, except when they are on vacation. Each pet has one employee from each shift who is booked as being primarily responsible for the pet from the start to the end of its stay, because this ensures consistency of care. At the end of any booking, we produce a report card in our system, which we can also print for the client."

Soyoon Kim, the central administrator, told us:

"I register all the pets. We need the name, of course, plus whether it is a dog or cat, and the colour and breed, which help the staff identify it. We also keep track of whether vaccine status is up-to-date but really that's just a yes or no, and if the no isn't corrected before the next visit, the pet can't board with us anymore. There's a care profile of feeding and medical instructions for each pet, and we store an emergency vet contact for each pet. For some reason, although we have thousands of pets in our system, there are only about a hundred vets that our clients use, so right now we keep the emergency vet's name and phone number in a spreadsheet. I think the reason that there are so few vets is that a lot of our clients have multiple pets, and they usually use the same vet for each pet - although I do know one person who has one vet for their dog,

and another for their cat. I really look forward to being able to see that information when I'm looking at a pet profile!"

Paul Asukwo, our lead requirements engineering analyst at ENSF409 Software, asked you, our junior analyst, to complete the UML diagram. He drew up a few notes to get you started. He mentioned that for this prototype, you can use a simplified approach for contact details as in his example below. He also mentioned that to keep things simple, for now everything can be made public in the code and access control will be worked out later.



ENSF409 Software's lead programmer, Mehvish Jaffary, started working on the prototype, but only had the chance to write a few stubs before she was needed on another project. She left the following notes.

```
class Pet{

    public String name;
    public String species;
    public String breed;
    public String colour;
    public boolean vaccine = false;
    public Client owner;
    public EmergVet vet;
    public CareProfile care;

    public Pet(String name, String species, String breed, String colour,
          Client owner){
    }

    public void setVet(String vetName, String vetNumber){
    }

    public void setCare(String[] medList, String medInstr, String feedInstr){
    }

    public void updateVaccineStatus(boolean changeStatus){
    }

    public boolean getVaccineStatus(){
    }
}

class Client{
}

class RewardsProfile{
}

class Booking{
}

class ReportCard{
}

class Employee{
}

class Store{
}

class CareProfile{
}

class EmergVet{
}
```