**Problem Statement/Scenario:**

**Monitoring System:**

I want to create a program that can monitor the animals for the zookeeper. This program needs to ask the user if they would like to monitor an animal or habitat, then go through the options of which animal or habitat, then display that information. I also would like the program to display a warning box if any of the details are out of range, like a cut on an animal, or food being low. This program needs to run with ease for the user and can pull information from the files on hand. This will make it easier for the zookeeper to know what is going on in the zoo right away rather than walking around and trying to figure out what is going on.

**Overall Process:**

The amount of times I changed this program was ridiculous. I couldn’t get things to work right and it was driving me nuts so I would start all over. The final program was a disaster for the longest time, but then I finally figured it out and it worked. I started off by making a loop so it can go back through the options repeatedly until the user exited the program. I added an integer ‘userChoice’ (choice) to equal the menu() method. I then created a switch statement so if the user entered 1, the program would open the animal.txt file, if the user entered 2 the program would open the habitats.txt file, and the 3 for the program to close.

**Pseudocode:**

My program somewhat matched my pseudocode, I had to change a few things. I did not have a loop in my pseudocode, and I realized that when we did the discussion post with our peers. I needed the program to loop for the program to loop back through the menu options. I added a switch statement within my loop as well. I completely forgot about that until you mentioned it in the help file and thank you so much because it made my program more easily read, at least to me. For the user input, I switched it to numbers instead of a word because it was easier to fix if the user inputted the wrong number, rather than the wrong word. It’s also simpler to the user rather than typing out ‘Animal’, ‘Lion’, ‘Exit’. Another thing I forgot in my pseudocode was to talk about putting in a txt file. I’m not sure if that necessarily needed to be in my pseudocode, but it’s something that changed and brought into my program. Same with the pop up message board for a dialogue box. I have put my changes in red on my pseudocode

method main() {

// I used a while loop to loop through all of the options repeatedly

// I used numbers instead of words for the ease of the user. Integers worked better than strings for me

// I made a menu() method rather than just printing a menu

DISPLAY options // Ask the user to enter the following options

* Monitor an animal
* Monitor a habitat
* Exit

// Use an if-else statement to have the program select the correct file

IF zookeeper = monitorAnimal // if the user wants to monitor an animal, they would select this option

Display the list of animals

* Lion
* Tiger
* Bear
* Giraffe

IF zookeepr = monitorHabitat // if the user wants to monitor a habitat, they would select this option

Display the list of habitats

* Penguin
* Bird
* Aquarium

ELSE zookeeper = Exit // if the user does not want to get any information, exit the program

ENDIF

// Use a while loop to go to the correct file depending on the users input

WHILE zookeeper = monitorAnimal

Prompt user on which animal

DISPLAY list of animals // Let user look at the options available to get information on

* Details on Lion
* Details of Tigers
* Details on Bears
* Details on Giraffes
* Exit

// ended up using a switch statement to pull from the .txt files vs displaying a menu each time

INPUT zookeepers response

// Create if-else statement to give the zookeeper the correct animal that is asked for

IF zookeeper = Lion

DISPLAY Lion details

ELSE IF zookeeper = Tiger

DISPLAY Tiger details

ELSE IF zookeeper = Bear

DISPLAY Bear details

ELSE IF zookeeper = Giraffe

DISPLAY Giraffe details

ELSE zookeeper = Exit

ENDIF

WHILE zookeeper = monitorHabitats

Prompt user on which habitat

DISPLAY list of habitats // Let user look at the options available to get information on

* Details on penguin
* Details on birds
* Details on aquarium
* Exit

// Create if-else statement to give the zookeeper the correct habitat that is asked for

IF zookeeper = Penguin // if the zookeeper types ‘Penguin’, then the program should display the penguin habitat details

DISPLAY Penguin habitat details

ELSE IF zookeeper = Bird // if the zookeeper types ‘Bird’, then the program should display the bird habitat details

DISPLAY Bird habitat details

ELSE IF zookeeper = Aquarium // if the zookeeper types ‘Aquarium’, then the program should display the aquarium habitat details

DISPLAY Aquarium habitat details

ELSE

ENDIF

**Methods and Classes:**

I did not specify any methods/classes when making the pseudocode. But technically it is a change in my documentation. At first I did not use your help file because I could not get anything to pull correctly. I emailed you and they pulled for you, but on my end the files refused to work unless I put them 3 levels above my .java files. I still have no idea why it’s doing that, but as long as it works I’ll go with it. Using methods/classes make coding a lot easier, because rather than retyping everything several times, I can just pull up an object and reuse it in multiple areas. I can just type one sometimes two words instead of System.out.println(..) a bunch of times.

**Error Documentation:**

The amount of errors was unbelievable if I’m honest. First of all, it took me some time to begin even reading the help file. It was not very understandable so I had to sit for a while and read it over. Then trying to figure out the error message itself to pop up, I had the popUP line in the completely wrong area of the code, don’t know why my brain put it in the spot it was in because it had nothing to do with that area of the code. It was very entertaining learning how my brain functions at a 1,000 mph and under stress. Then I watched the video from the announcements and he explained the order it needed to be in and that helped tremendously. Another error was I was indecisive on whether I wanted the user to enter a word or number, so I had a mixture on integers and strings. Which was really fun to fix throughout the coding process. All in all, it was fun trying to make this code perfect and working functionally. Had a great laugh at some of the errors I made. My major problem though throughout this entire process was trying to pull information off of the txt files. For me, NetBeans would say it was an ‘Unrecognized File’ if it was next to my .java files. I had to put it 3 levels above my java files in order for it to work for me. I turned this project in hoping it would work for you if the txt files were next to the java files. I will have no clue if they did until you grade the project… Which makes me nervous, but we will have to see.

**Solution Documentation:**

Fixing the errors was the worst part of this coding project. I made the goofiest mistakes. I had integers mixed with strings and strings mixed with integers, I had the popup warning under the opening file area. I had to go make changes to my values to either be a string or an integer, not both, which I had to go all throughout the code to make sure it was all lined up and matching. Then the popup message had to be moved down to the error message area because that’s where it should have been that entire time. I learned that my brain puts things in places that shouldn’t be and that I over read things. I’ll think a line is correct and skip over it and then later realize that line was the problem the entire time. It was frustrating and entertaining at the least. Also moving the txt files all over the folder was fun to see where it would work or not. But mainly my eyes would skip over lines and those lines would be the biggest problem in my code.