**Chapter 5**

**Need help with the addSigthings method.**

Ex. 5.1 – Done

Ex 5.2 - Done

Ex 5.3 – Done

Ex. 5.4 – In many cases for loops and while loops can be used interchangeably. The reason one is chosen over the other is typically simply based on what we want to accomplish with it.

Ex. 5.5 – Rewrite the prinList method in animal-monitering-v1 using a lambda.

Done and Saved

Ex. 5.6 – Done and saved

Ex. 5.7 –Write pseudo code to determine how many elephants a spotter saw on a particular day.

Sightings.filter( name == elephant).filter(spotter == spotterID). Filter (day == dayID). map(count).reduce(add)

Ex. 5.8 – Write pseudo code to create a steam containing only those sightings that have a greater count than 0

Sightings.map(count).filter(count > 0)

Ex. 5.9 – Write pseudo code to determine the total number of times tracks by the artist “Big Bill Broonzy” have been played.

MusicOrginizer.filter(artistName == Big Bill Boozy).map(count).reduce(add)

Ex. 5.10 – Rewrite the printSightingsOf method in the animal monitor class use streams and lambdas.

Done and saved

Ex. 5.11 – Write a method in AnimalMonitor to print details of all sightings recorded on a particular dayID.

Done and saved

Ex. 5.12 – Write a method that uses two fileter calls to print details of the sightings of a particular animal made on a particular day.

Done and Saved

Ex. 5.13– The order of the filter calls doesn’t matter. In the way that I did it filters all sightings of a certain animal and then filters again for a specific period of the sighing. If we did it the other wat the method would filter out all sightings on everything but the specified day and then filter again for the desired animal sightings on that day. Both work if written correctly.

Ex. 5.14– Rewrite the printSightingsBy method in your project as discussed above.

Done and Saved.

Ex. 5.15– Write a method to print the counts of all sightings of a particular animal.

Done and Saved.

Ex. 5.16 – Yes if a pipeline contains both a map and a filter operations the order matters if you want to get the desired result. This is because map operations take all information from the original stream and maps it to a new place in a new stream. The filter however takes a selected element and creates a new stream with the selected items.

Ex. 5.17 – Rewrite the printEndangered method in your project to use streams.

Done and Saved.

Ex. 5.18 – Change the printSightingsBy method

Done and saved

Ex. 5.19– Having issue with final return

Ex. 5.20– Add a method to AnimalMonitor that takes three parameters: animal, spotterID, and dayID, and returns a count of how many sightings of the given animal were made by the spotter on a particular day.

Having issue with final return

Ex. 5.21– Write a method that takes two parameters, spotterID and day, and returns a string containing the names of the animals seen by the spotter on a particular day.

Check Functionality

Ex. 5.22– Re-write the removeZeroCounts method using the removeIf method.

Done and saved.

Ex. 5.23– Write a method removeSpotter that removes all records by a given spotter.

Ex. 5.24– Done

Ex. 5.25– Write a method in the AnimalMonitor class that takes a single parameter spotterID and returns a count of how many sightings have been made by the spotter.

Ex. 5.26– Write a method that takes an animal name as a parameter and returns the largest count for that animal in a single Sighting record.

Ex. 5.27– Write a method that takes an animal name and spotterID and returns the first Sighting object stored in the sightings collection for that combination.

Can get the method header

Ex. 5.28–

The limit method w

Ex. 5.29– Use streams or lambdas to re-write the listAllTracks and ListByArtist method in the MusicOrganizer class.

Done and saved.

Assignment 4 project –

**Web Site Sales by Lame-O**

You are a programmer for Lame-O web design company. Currently your company offers five different 'cookie-cutter' designs that customers can select from. Your job is to create a sales application for your company’s web site so that customers can purchase their own designs. Once they have selected a design, they can choose from five unique features in each design. Each feature adds to the cost of their site. See the table below for the designs and added costs:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Design | Base Price | Feature 1 | Feature 2 | Feature 3 | Feature 4 | Feature 5 |
| Nature | 300 | 10 | 15 | 20 | 25 | 30 |
| Tech | 350 | 20 | 30 | 40 | 50 | 60 |
| Business | 375 | 30 | 40 | 50 | 60 | 70 |
| Music | 400 | 85 | 95 | 110 | 130 | 210 |
| Naughty | 500 | 100 | 200 | 300 | 400 | 500 |

* Write a program that automatically creates a unique 5-digit customer number: DONE
* Store customer data: Last & first name, company name, address, city, state, zip, phone: TWEAK
* Allow customer to choose design type and have features with prices displayed: IN PROGRESS
* Allow customer to select design and features: IN PROGRESS
* Geneterate a bill of sale for customer: FINAL METHOD TO BE USED
* Store customer data & order information: IN WHAT AND HOW