

**Developer**: Jessica Ayer

**Date**: 07/25/2021

# IT 145 Global Rain Summary Report Template

## Pseudocode

-CREATE Pet Class to get/set pet type, pet name, pet age, cat spaces, dog spaces, days staying, and amount due.

-INITIATE variables for price of cat stay/day, price of dog stay/day/wight category and grooming price/weight category.

-GET user input for pet Info.

-CREATE an array for cat spaces 1-12

-CREATE an array for dog spaces 1-30

* WHILE user input is not quit:
  + IF the pet is a cat:
    - IF there is an empty space in the cat spaces array:
      * ADD the cat to the cat spaces array.
      * MULTIPY the length of stay by the price of cat stay/day.
      * OUTPUT the amount due.
    - ELSE:
      * OUTPUT that there are no spaces available for the cat.
  + IF pet is a dog:
    - GET the dog’s weight.
    - GET if the dog requires grooming.
    - IF there is an empty space in the dog array:
      * ADD dog to the dog spaces array.
      * IF no grooming is required:
        + If the Dog is greater or equal to 30lbs:

MULITPLY the length of stay for by the price of large dog stay/day.

OUTPUT Amount due.

* + - * + ELIf the Dog is less than 30lbs and greater than or equal to 20lbs:

MULITPLY the length of stay for by the price of medium dog stay/day.

OUTPUT Amount due.

* + - * + ELIF the Dog is less than 20lbs:

MULITPLY the length of stay for by the price of small dog stay/day.

OUTPUT Amount due.

* + - * ELSE: //dog requires grooming
        + IF the Dog is greater or equal to 30lbs:

MULITPLY the length of stay for by the price of large dog stay/day and add the large dog grooming fee.

OUTPUT Amount due.

* + - * + ELIF the Dog is less than 30lbs and greater than or equal to 20lbs:

MULITPLY the length of stay for by the price of medium dog stay/day and add the medium dog grooming fee.

OUTPUT Amount due.

* + - * + ELIF the Dog is less than 20lbs:

MULITPLY the length of stay for by the price of small dog stay/day and add the small dog grooming fee.

OUTPUT Amount due.

* + - ELSE:
      * OUTPUT that there are no spaces available in the dog.
  + ELSE:
    - IF the user inputs “quit”
      * EXIT the program.

## Flowchart

**Is** **space for dog?**

yes

**Get dogs weight and assign space number.**

**IF Dog**

yes

no

no

**Grooming?**

**Output here is** **no space.**

**Is there space for cat?**

**ELIF Cat**

yes

yes

**ELSE assign cat space**

yes

no

no

**Elif Quit**

**Output here is no space.**

**Output amount due**

## OOP Principles Explanation

For this project, all four OOP principles will be implemented. The main class members will all be **encapsulated** so that Global Rain can ensure the program is used as intended. Get/Set methods will be implemented so that the original variables are never changed and just used within the correct program methods. **Inheritance** will be used for creating each pet, or client’s profile so that we can ensure all of the correct data is collected and referenced by the correct methods within the program. This will tie into **polymorphism** is what takes place when the user inputs each individual pets’ information into the system. Each pet profile will be created through the use of the same class member but will contain unique information and parameters that are pet specific. These unique qualities are things like the type of pet, the pet’s name, the pet’s weight, how many days the pet will be boarded and so forth.

**Abstraction** will be used so to make the final product user friendly. There is a lot of background coding that will come in to play to set up each variable, calculate prices, and process methods but the user only needs see the prompts for input information and an easy-to-read output.