

**Developer**: Shenika Eayrs

**Date**: 3/19/2023

# IT 145 Global Rain Summary Report Template

## Directions

Place your pseudocode, flowchart, and explanation in the following sections. Before you submit your report, remove all bracketed text.

## Pseudocode

// The Problem:

//Automate Pet Boarding and Grooming Check-in

// Gather Pet details and Stay Duration from User

// Validate available spaces, assign pet a space ID and provide a tentative amount due

START

INPUT – Greet Client confirm type of Pet [petType] and confirm if booking for [newpet] or [returningpet]

**\*\*\*[returningpet]\*\*\***

INPUT - IF [Y] PROMPT User to confirm if there are changes to [Petprofile] (Y or N)

INPUT - IF [Y] update [Petprofile] and proceed

INPUT - IF [N] confirm the stay duration ( [getDaysStay]) and proceed

GET – search for available spaces [getcatSpaces] or [getdogSpaces]

RETURN- return available spaces [return.catSpaces] or [return.dogSpaces]

OUTPUT IF no space available for [PetType] i.e. (getDogSpaces() <= 0) advise User and END Program

INPUT - IF [PetType] = dog AND if(dogdur >= 2){PROMPT User to confirm the need for grooming

Save INPUT to [PetProfile]

ASSIGN pet a [SpaceID]

UPDATE available spaces [setcatSpaces] or [setdogSpaces]

SAVE Booking to [PetProfile]

**\*\*\*[newpet]\*\*\***

INPUT - PROMPT User to confirm Pet Details

CREATE [PetAcct] using the [PetType], [PetName], [PetAge] User defined

INPUT - confirm the stay duration ( [getDaysStay]) and proceed

GET – search for available spaces [getcatSpaces] or [getdogSpaces]

RETURN- return available spaces [return.catSpaces] or [return.dogSpaces]

OUTPUT IF no space available for [PetType] i.e. (getDogSpaces() <= 0) advise User and END Program

INPUT - IF [PetType] = dog AND if(dogdur >= 2){PROMPT User to confirm the need for grooming

Save INPUT to [PetProfile]

ASSIGN pet a [SpaceID]

UPDATE available spaces [setcatSpaces] or [setdogSpaces]

SAVE Booking to [PetProfile]

END

## Flowchart



## OOP Principles Explanation

The OOP principles are synergized and facilitate an ergonomic application that provides a programmer with transformable sustainable code, which allows a class or function to be modified without affecting the other parts of an application. For the Semester Pet Bag project, the Encapsulation task was leveraged to define a Pet Class. In addition, the Polymorphism phase was used to permit the Pet class to behave differently depending on the invoking object types (i.e., petType, catSpaces, or dogSpaces). Lastly inheritance was used to develop a subclass for Pet from the Parent Pet class. The assessor properties and methods were seamlessly inherited from the Parent class to the subclass.