

**Developer**: Victoria Kaloudis

**Date**: 25July2021

# 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

When you are done implementing the Pet class, refer back to the Pet BAG Specification Document and select either the pet check in or check out method. These methods are detailed in the Functionality section of the specification document.

Write pseudocode that lays out a plan for the method you chose, ensuring that you organize each step in a logical manner. Remember, you will *not* be creating the actual code for the method. You do *not* have to write pseudocode for both methods. Your pseudocode must not exceed one page.

START with Pet Check in

IF pet == dog THEN determine available dog spaces

IF available dog spaces >= 0 THEN determine stay length

IF stay length >= 2 THEN determine need for grooming

IF grooming THEN make appointment for grooming

ELSE obtain information on pet and save data {

Get petName

Get petAge

Get petWeight

Get stayLength

}

ELSE obtain information on pet and save data {

Get petName

Get petAge

Get petWeight

Get stayLength

}

ELSE no services are available

EL IF pet == cat THEN determine available cat spaces

IF available cat spaces >= 0 THEN obtain information on pet and save data {

Get petName

Get petAge

Get petWeight

Get stayLength

}

ELSE no services are available

ELSE no services are available

## Flowchart

Based on the pseudocode you wrote, create a flowchart using a tool of your choice for the method you selected. In your flowchart, be sure to include start and end points and appropriate decision branching, and align the flowchart to the check in or check out process. Your flowchart must be confined to one page.

‘Graphical user interface, application, Word

Description automatically generated

**Pet Check Out**

START with Pet Check Out

IF pet == dog THEN determine dog weight

IF dog weight >= 30 pounds THEN calculate large weight fee

IF dog weight <30 pounds THEN calculate small weight fee

IF dog == groomed THEN determine grooming fee

IF dog weight >= 30 pounds THEN calculate large weight fee plus grooming fee

IF dog weight <30 pounds THEN calculate small weight fee plus grooming fee

IF dog += groomed THEN calculate weight fee

IF dog weight >= 30 pounds THEN calculate large weight fee

IF dog weight <30 pounds THEN calculate small weight fee

Calculate total fees

Mark dog space as vacant;

ELIF pet == cat THEN determine days stayed

Calculate fees for stay

## OOP Principles Explanation

Briefly explain how you applied object-oriented programming principles and concepts (such as encapsulation, inheritance, and so on) in your software development work thus far. Your explanation should be one paragraph, or four to six sentences.

The four main pillars of OOP are inheritance, encapsulation, abstraction, and polymorphism. Inheritance of an object is the ability of the object to acquire properties of a different object. We extended sub classes from a superclass using inheritance. We extended characteristics and onto objects of another class. Encapsulation is when an object has a private state in its class. All classes programmed has private integers and strings, such as the Pet Name and Pet Age. Abstraction works as an extension to encapsulation where the object has public functions and can only access the public functions. The public functions work as getters and setters where we can set the characteristic and return the value of the characteristic. Finally, polymorphism allows a larger class to have little classes within that keep the same methods. We were able to modify the sub classes within the superclass to keep the same methods, just applying to the sub classes. Using these pillars, programmers can code more easily, more simply, and safely. Programmers are easily able to reuse code and can then code faster by creating classes. It can also be easily copied and pasted and then updated to allow for faster coding.