

**Developer**: Brian Engel

**Date**: 11/17/2022

# 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.

Pet check in

setPetType

if getPetType is dog then

if getDogSpaces is greater than 0 then

setPetName

setPetAge

setDaysStay

setDogWeight

setDogSpaceNumber

setDogSpaces to 1 less

if getDaysStay is equal to or greater than 2 then

setGrooming

else (there is no space)

print there is no space

else if getPetType is cat then

if getCatspaces is greater than 0 then

setPetName

setPetAge

setDaysStay

setCatSpaceNumber

setCatSpaces to 1 less

else (there is no space)

print there is no space

else (it’s not a dog or cat)

print invalid animal

## 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.

Diagram

Description automatically generated with medium confidence

## 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 program applies object-oriented programming principles by using private variable in the class Pet which is encapsulation. It uses abstraction by only having constructors, mutators and accessors messing with the “guts” of the class. There is inheritance because you can make more than 1 instance of the Pet class and each will have the same set of variables and methods, but they can hold different values. Lastly, it has polymorphism because the Cat and Dog classes will be an extension of the Pet class, and build on the methods and variables already established there.