

**Developer**: Angel Davila

**Date**: 7/23/2022

# 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.

Note: Likely in the main function we would set some sort of state for our store, which would track transactions, pet spaces, etc. I would suggest we move have a new class that will handle state, and we can initialize it as a singleton object right when the application starts.

// determine the type of pet

FUNCTION FindPetType(petype, petInfo) -> Pet:

SWITCH petType:

CASE Dog:

CREATE Dog(petInfo)

RETURN Dog object

CASE Cat:

CREATE Cat(petInfo)

RETURN Cat object

// this function will check the number of spaces we have and if we are allowed to add pet

// this function will reach out to the state

FUNCTION CheckSpaces(pet) -> boolean:

SWITCH petType:

CASE Dog:

CHECK state for dog capacity

RETURN true or false

CASE Cat:

CHECK state for dog capacity

RETURN true or false

// we can gather all the pet info before this function, we want to make the function as bear bone as possible

// this is why the PetCheckIn function only handles storing the pet. Pet info is all the information we need.

FUNCTION PetCheckIn(petType, petInfo) -> boolean:

VAR pet = FindPetType(petType, petInfo):

VAR spaces = CheckSpaces(pet)

IF !spaces:

RETURN “ No spaces”

ELSE store pet into state

RETURN true or false

Note: We could remove CheckSpaces() and perform this action before we gather the pet information just to be more efficient. For now we can write this and optimize later.

FUNCTION GetGroomingFee(pet) -> integer:

INTEGER amount

IF pet.getGrooming()

// we can add a new method on the dog object to calculate grooming fee

amount += pet.getGroomingFee()

// we can add the total boarding fee here

// might want to even combine this line and the above line into one method

amount += pet.getBoardingFee()

RETURN amount

FUNCTION PetCheckOut(pet) -> Integer:

CHECK IF pet exists

VAR amount = GetGroomingFee()

REMOVE pet object from state

REMOVE pet space from state

RETURN amount

## 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.

Main Function

Ask for pet info

Check spaces available

Get pet information

Check in, add pet to state

Checkout remove pet from state

## 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 important part of OOP is to be able to abstract code away from the developer to speed up development time. This code can be used in multiple places in the current application and even in outside applications. Here we encapsulate all the data about the pet into a main Pet object and encapsulate the logic that will be performed for this specific pet inside of methods. This way we can create an object, call these methods from anywhere in our application. Also Pet will be the parent object and Cat and Dog will inherit, this allows us share date between both object types and even return both in a function, since they are of the same parent.