

**Developer**: Armon Wilson

**Date**: 5/28/2023

# 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

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.

GET petType

GET petType space availability

IF petType space available

IF returning pet

GET petName

GET and SET petAge, petWeight

ELSE

SET petType

GET and SET petName, petAge, petWeight

GET and SET daysStay

IF daysStay is greater than 2 and petType is Dog

GET and SET grooming (yes or no)

GET and SET spaceNumber

SUBTRACT 1 from space availability for petType

OUTPUT spaceNumber

End Check-in

ELSE

OUTPUT “No Space Available”

End Check-in

## 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.

**A diagram of a pet

Description automatically generated with low 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.

1. Encapsulation:
   * The attributes (petType, petName, etc.) are declared as private, encapsulating them within the Pet class and restricting direct access from outside the class.
   * Public getter and setter methods (getPetType, setPetType, etc.) are provided to control access to these attributes and enforce data encapsulation.
2. Inheritance:
   * The code consists of subclasses like Dog and Cat. The dogSpaces and catSpaces attributes inherit from the Pet class, inheriting its attributes and behaviors.
   * The Dog.getDogSpaceNumber() and Cat.getCatSpaceNumber() methods are called, these subclasses provide their own implementations for space numbers.
3. Abstraction:
   * The Pet class serves as a general abstraction for pets, with attributes and methods that can be inherited or overridden by more specific pet types (e.g., dogs or cats).
   * The class methods (getPetType, setPetType, etc.) provide a higher-level interface for interacting with pet attributes, hiding the underlying implementation details.