

**Developer**: Maridelle Anne J. Gonzales

**Date**: March 31, 2024

# IT 145 Global Rain Summary Report Template

## 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 Method Pseudocode*

// Determine if the pet is a dog or a cat and check for boarding space.

Print("Is the pet a dog or a cat?")

Read petType

IF petType is = "dog"

IF dog\_Space is less than 30

Assign dog\_spacenumber

Decrease dog\_Space by 1

ELSE:

Pinrt("Sorry, no dog boarding space available.")

Terminate the check-in process

ELSE IF petType is = "cat"

IF cat\_spaces is less than 12

Assign cat\_spacenumber

Decrease cat\_spaces by 1

ELSE:

Print("Sorry, no cat boarding space available.")

Terminate the check-in process

ELSE

Display "Invalid pet type. Please choose 'dog' or 'cat'."

Terminate the check-in process

// Collect the appropriate information

Print("Enter pet name:")

Read petName

Print("Enter pet age:")

Read petAge

IF petType is = "dog"

Print("Enter pet weight:")

Read dog\_weight

// Gather information on the length of stay

Print("Enter length of stay in days:")

Read daysStay

// Determine if the pet owner wants grooming (only for dogs staying two or more days)

IF petType is = "dog" AND daysStay >= 2

Print("Does the pet owner want grooming? (yes/no)")

Read grooming\_choice

IF grooming\_choice = "yes"

Set grooming to true

ELSE:

Set grooming to false

// Assign the pet to a space

IF petType = "dog"

Assign petType, petName, petAge, dog\_spacenumber, daysStay, grooming to Dog class

ELSE IF petType = "cat"

Assign petType, petName, petAge, cat\_spacenumber, daysStay to Cat class

Print("Pet check-in successful. Assigned space: [assigned space number]")

## Flowchart

A diagram of a dog

Description automatically generatedBased 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.

## 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.

In my software development work such as developing the pseudocode for the pet check-in process, I applied several object-oriented programming principles. Encapsulation was evident in the organization of the code, where related data and functionality were encapsulated within classes such as Dog and Cat. Inheritance was utilized implicitly in the design, as both Dog and Cat classes could inherit common attributes and methods from a parent class representing a generic Pet. Moreover, the pseudocode demonstrated abstraction by focusing on the high-level functionality of the check-in process, hiding the complexities of implementation details. Polymorphism played a crucial role, enabling flexibility in method invocation based on the type of object, promoting modular and extensible code. Additionally, I've leveraged interfaces to define contracts for classes, promoting loose coupling and facilitating dependency injection, thus enhancing the scalability and testability of the software. Overall, embracing OOP concepts has enabled me to create robust, modular, and maintainable software solutions.