

**Developer**: Tatiana Epps

**Date**: November 24, 2024

# 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

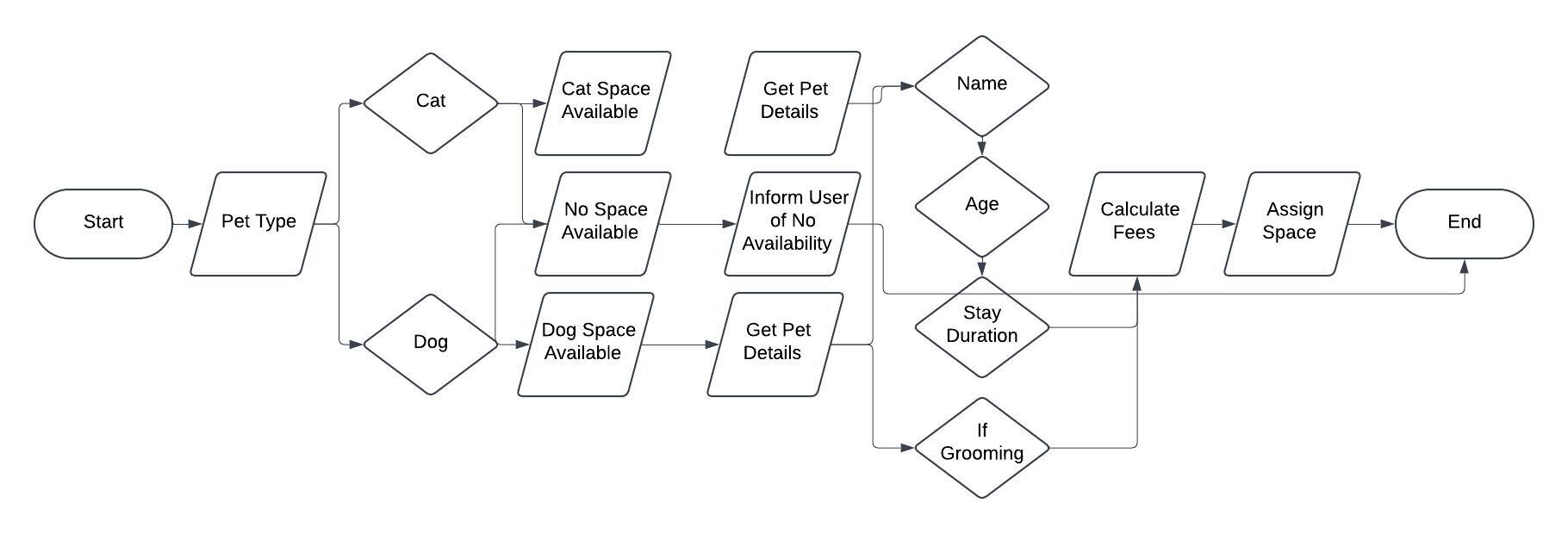
When you are done implementing the Pet class, refer back to the Pet BAG Specification Document and select either the pet check in or check out method. These methods are detailed in the Functionality section of the specification document.

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.

1. START
2. Prompt user for pet type (dog or cat)
3. If dog:
   1. Check available dog spaces (getDogSpaces())
   2. If dog spaces > 0:
   3. Proceed with check-in process
4. Else:
   1. Inform user no dog spaces available
5. Else if cat:
   1. Check available cat spaces (getCatSpaces())
6. If cat spaces > 0:
   1. Proceed with check-in process
7. Else:
   1. Inform user no cat spaces available
8. Check-In Process
   1. Get pet name (setPetName())
   2. Get pet age (setPetAge())
   3. Get length of stay (setDaysStay())
   4. Determine grooming eligibility (dog && daysStay >= 2)
   5. If eligible, prompt user for grooming choice (setGrooming())
   6. Calculate potential amount due (based on pet type and days)
   7. Assign pet to a space (update dogSpaces or catSpaces)
9. END

## 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.



## 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.

This initial design incorporates Object-Oriented Programming (OOP) principles through encapsulation and inheritance (future implementation). The Pet class encapsulates the data (attributes) and behaviors (methods) related to a pet object. This promotes data protection and controlled access through accessors (getters) and mutators (setters). While not fully implemented here, the design allows for future inheritance. We can create Dog and Cat subclasses that inherit attributes and methods from the Pet parent class. This promotes code reusability and reduces redundancy.