

**Developer**: DeMarko Stephens

**Date**: 5/28/2024

# 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.

Pseudocode for the pet check-out method.

START

IDENTIFY the pet and its boarding space

IF pet is a Dog AND stayed for 2 or more days THEN

IF dogWeight >= 30 lbs THEN

SET groomingFee to 29.95

ELSE IF dogWeight >= 20 AND dogWeight < 30 THEN

SET groomingFee to 24.95

ELSE IF dogWeight < 20 THEN

SET groomingFee to 19.95

END IF

ELSE

SET groomingFee to 0

END IF

IF pet is a Dog THEN

IF dogWeight >= 30 lbs THEN

SET boardingFee to daysStay \* 34.00

ELSE IF dogWeight >= 20 AND dogWeight < 30 THEN

SET boardingFee to daysStay \* 29.00

ELSE IF dogWeight < 20 THEN

SET boardingFee to daysStay \* 24.00

END IF

ELSE IF pet is a Cat THEN

SET boardingFee to daysStay \* 18.00

END IF

CALCULATE totalFee as boardingFee + groomingFee

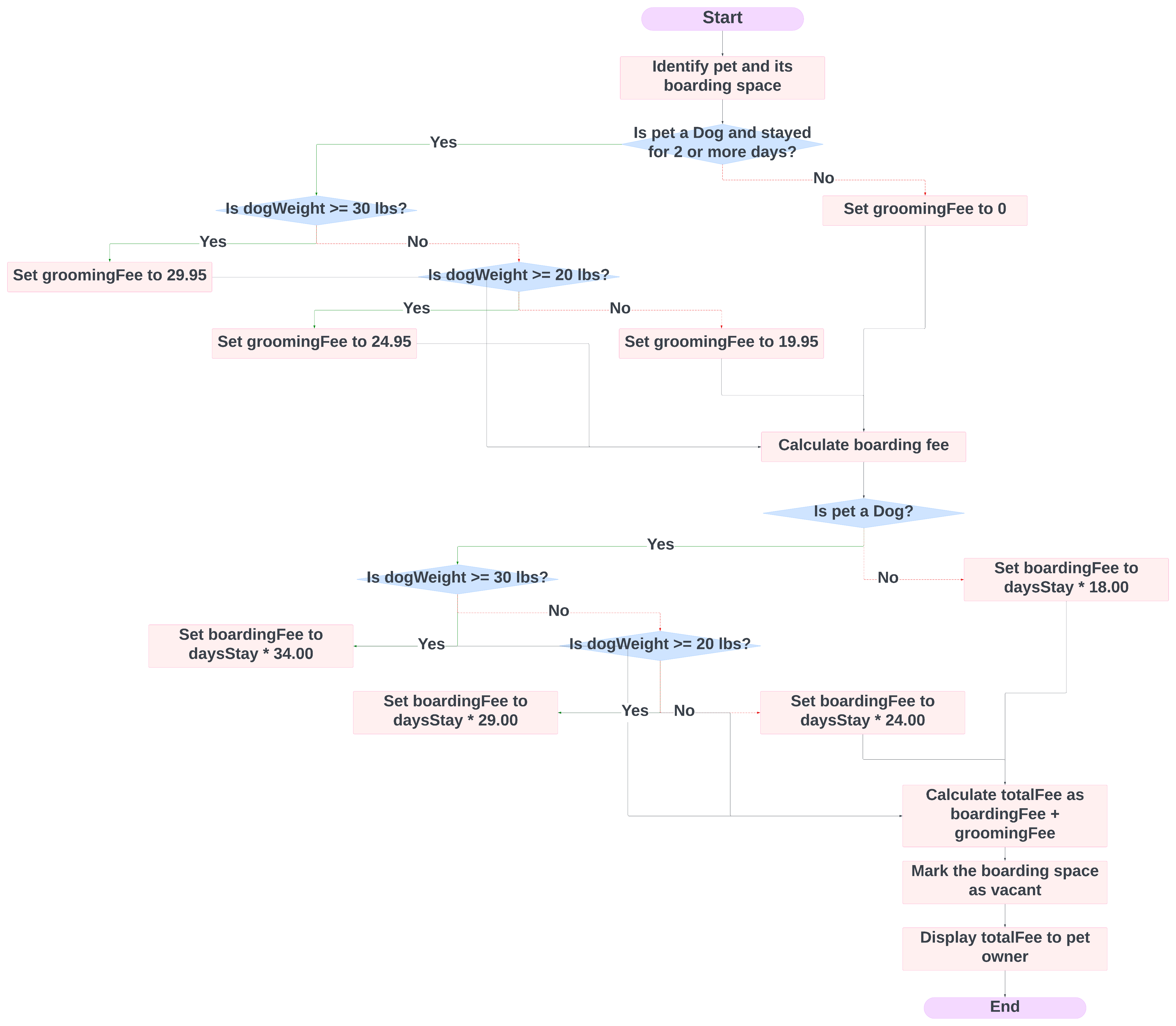
MARK the boarding space as vacant

DISPLAY totalFee to pet owner

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.

In the creation of the Pet class, the concepts of encapsulation and inheritance were employed to enhance the design. Encapsulation is evident through the implementation of private variables coupled with public getters and setters, safeguarding the object’s state and ensuring controlled access, which bolsters data integrity and security. Inheritance is suggested by the UML diagram, indicating a hierarchical connection among the Pet, Dog, and Cat classes. This arrangement allows the Dog and Cat classes to derive common attributes and methods from the Pet class, facilitating code reuse and the orderly grouping of related entities. Adhering to these fundamental OOP principles is instrumental in developing software that is both flexible and robust.