

**Developer**: Daniel Guerrero

**Date**: 09/24/2023

# 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.

Start

Pet Check-In

Input pet type (dog or cat)

If pet type is dog:

Check if space is available (Max 30)

If space is not available:

Output “No space available” and End

Else:

Prompt for new or returning dog

If user chooses new dog:

Collect dog information

Else if user chooses returning dog:

Update dog information

Prompt user input for length of stay

If length of stay 2 or more days:

Prompt for grooming (yes or no)

Assign pet location

Decrease available dog space by 1

If pet type is cat:

Check if space is available for cats (Max 12)

If space is not available:

Output “No space available” and End

Else:

Prompt for new or retuning cat

If user chooses new cat:

Collect cat information

Else if user chooses returning cat:

Update cat information

Prompt user input for length of stay

Assign pet location

Decrease available cat space by 1

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.

A diagram of a cat

Description automatically generated

## 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.

Learning about the OOP principles has been a major help in creating my code. In my work so far, I have been able to use encapsulation by restricting access to public methods, ensuring data integrity by using private instance variables, and providing controlled access through public accessor methods. For abstraction, I’ve separated my code into distinct classes and methods that only request and manipulate information relevant to specific animals, promoting a clear and focused design. Polymorphism is also important here because both the Cat & Dog classes are closely related and can use some of the same methods while still being able to have a specific method for each if necessary. For example, dogs are consistently offered grooming if their stay is longer than 2 days, while cats, following the same abstraction principle, will never be offered grooming regardless of stay length. In the next few weeks, I look forward to learning how everything will tie together.

Pet.java Class Check-In

*/\*\**

*\*Pet class contains attrributes such ass pet type, pet name, pet age,*

*\*days of stay, space required, and amount due.*

*\*/*

public class Pet {

    private String petType;

    private String petName;

    private int petAge;

    private int daysStay;

    private int space;

    private double amountDue;

*// Constructors*

    public Pet(String petType, String petName, int petAge, int daysStay) {

        this.petType = petType;

        this.petName = petName;

        this.petAge = petAge;

        this.daysStay = daysStay;

    }

*// Getters and setters for Pet attributes*

*//Get type of pet*

    public String getPetType() {

*return* petType;

    }

*//Set type of pet*

    public void setPetType(String petType) {

        this.petType = petType;

    }

*//Get pet name*

    public String getPetName() {

*return* petName;

    }

*//Set pet Name*

    public void setPetName(String petName) {

        this.petName = petName;

    }

*//Get pet age*

    public int getPetAge() {

*return* petAge;

    }

*//Set pet age*

    public void setPetAge(int petAge) {

        this.petAge = petAge;

    }

*//Get days for stays*

    public int getDaysStay() {

*return* daysStay;

    }

*//Set days for stays*

    public void setDaysStay(int daysStay) {

        this.daysStay = daysStay;

    }

*//Get available space*

    public int getSpace() {

*return* space;

    }

*//Set available space*

    public void setSpace(int space) {

        this.space = space;

    }

*//Get amount due*

    public double getAmountDue() {

*return* amountDue;

    }

*//Set amount due*

    public void setAmountDue(double amountDue) {

        this.amountDue = amountDue;

    }

}

*//Dog class extends the Pet class and answers specific questions for Dog.*

class Dog extends Pet {

    private int dogSpaceNbr;

    private double dogWeight;

    private boolean grooming;

    public Dog(String petType, String petName, int petAge, int daysStay, int dogSpaceNbr, double dogWeight, boolean grooming) {

        super(petType, petName, petAge, daysStay);

        this.dogSpaceNbr = dogSpaceNbr;

        this.dogWeight = dogWeight;

        this.grooming = grooming;

    }

*// Getters and setters for Dog-specific attributes*

    public int getDogSpaceNbr() {

*return* dogSpaceNbr;

    }

    public void setDogSpaceNbr(int dogSpaceNbr) {

        this.dogSpaceNbr = dogSpaceNbr;

    }

*//Get dog weight*

    public double getDogWeight() {

*return* dogWeight;

    }

*//Set dog weight*

    public void setDogWeight(double dogWeight) {

        this.dogWeight = dogWeight;

    }

*//Grooming or no grooming*

    public boolean isGrooming() {

*return* grooming;

    }

    public void setGrooming(boolean grooming) {

        this.grooming = grooming;

    }

}

public class Main {

    public static void main(String[] args) {

    }

}