## Global Rain Logo

**Developer**: Joshua Langer

**Date**: May 24, 2020

# 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.

|  |
| --- |
| //Variables for Method  //petType  //petName  //petAge  //dogSpace  //catSpace  //daysStay  //amountDue    //Create checkIn() method  //Request Input from user for the pet type.  //Set petType to input from user check against scanner next and check against a boolean  //If petType is dog get dogSpace for empty space  //loop to assign dogSpace for pet and set dogSpace with petName, petAge, daysStay ->dogSpaceNum  //setGrooming()  //setDogWeight()  //calculate amountDue  //Else If petType is cat get catSpace for empty space  //loop to assign catSpace for pet and set catSpace with petName, petAge, daysStay ->catSpaceNum  //calculate amountDue |

## 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/check-out process. Your flowchart must be confined to one page.

|  |
| --- |
|  |

## OOP Principles Explanation

Briefly explain how you applied object-oriented programming principles in the software development process. Your explanation should be one paragraph, or four to six sentences.

|  |
| --- |
| I was able to apply OOP Principles to this project by applying correct Abstraction, Encapsulation, Inheritance and Polymorphism to the class. I kept the code easy to read and understand and applied proper accessors and mutators to ensure that the information coming from the user was set correctly so it could be retrieved properly on retrieval calls. I encapsulated the data in a constructor class to allow other classes to quickly access the information needed from other processing classes. I was able to set proper inheritance using static data flow. I was also able to apply polymorphism in a way that when a user calls a specific function from the application the variables would adjust accordingly based on the information the user wants to input or access. |