Competency

In this project, you will demonstrate your mastery of the following competency:

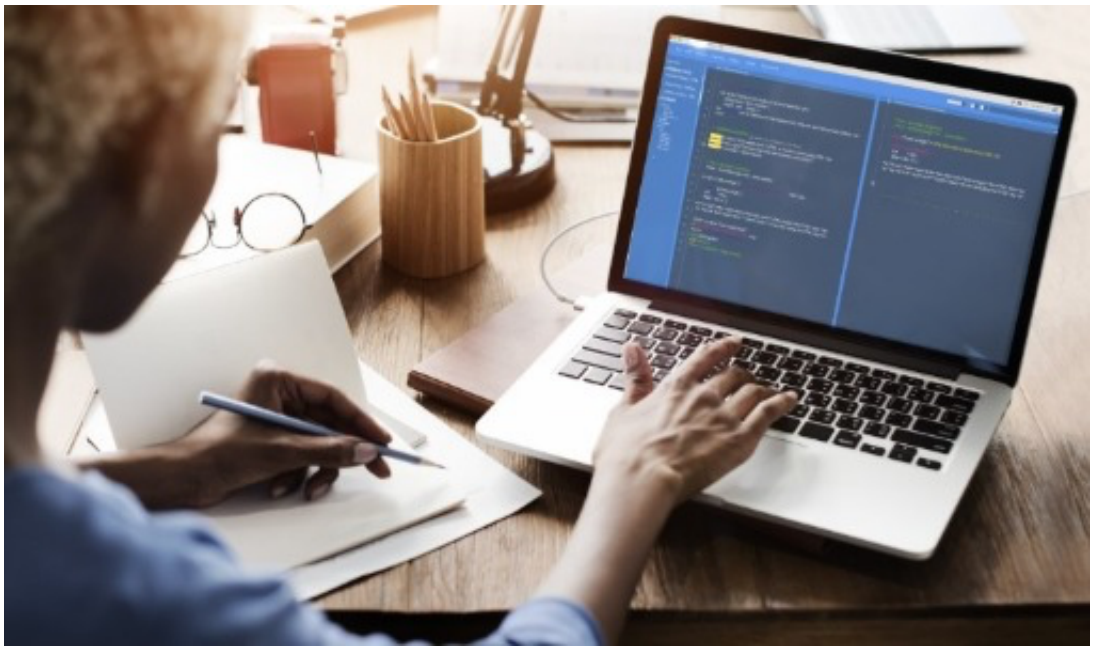
- Write programs by applying concepts and principles of object-oriented programming

Scenario

You work for Global Rain, a software engineering company that specializes in custom software design and development. As a junior software developer, you are part of a software development team that collaborates to create solutions for entrepreneurs, businesses, and government agencies around the world.

As part of a development team at Global Rain, you will be designing and developing custom software for a local pet boarding and grooming business, Pet Boarding and Grooming (Pet BAG). Pet BAG is seeking a way to modernize its operations by introducing custom software that will help replace inefficient paper-based processes. Your Global Rain team will collaborate to develop an app that will have the following initial functional areas: pet check in and pet check out.

As part of this team, you have been tasked with completing some preliminary work that consists of writing a Java class, writing pseudocode, and creating a flowchart. You will provide these deliverables to your supervisor.



Directions

Your supervisor has given you a specification document which details Pet BAG's software needs and includes a UML Class diagram. Using these documents, you will create both a Java class and a summary report. Your summary report will include pseudocode and a flowchart for *one* method (pet check in *or* pet check out), and an explanation of how your work meets object-oriented principles.

Pet.java Class File

1. Before you begin, it is important to understand what your client needs the software to do, and what work has already been done by your team. Review the Pet BAG Specification Document, located in the Supporting Materials section, which includes a UML Class diagram. Pay close attention to the class hierarchy, attributes, and behaviors.
2. To begin, open the Virtual Lab by clicking on the link in the Virtual Lab Access module. Then open your integrated development environment (IDE) and **create the Pet class based on the specifications in the UML Class diagram**. The Pet class must include the following:
    - **All attributes with appropriate data structures**. Note that the types are not specified in this UML class diagram. You will need to think about what the most appropriate data type is for each attribute.
    - **At least one constructor method**. You may use a default constructor. To score "exemplary" on this criterion, your constructor must initialize values for the petType, petName, petAge, and daysStay attributes.
    - **Accessors and mutators for all attributes**

    Note: You are *only* being asked to create one class. Though the UML diagram shows the Pet class as part of a larger application, for this project, you are not required to connect it to other classes or to try and run it as a program. Instead, you are practicing the skill of creating a class from a UML Class diagram.

Global Rain Summary Report

1. 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.
2. Open the summary report template, located in the What to Submit section. In the template, **write pseudocode that lays out a plan for the method you chose. Ensure that you organize each step in a logical manner and that your method meets the specifications in the document for either the check in or check out process**. Your pseudocode must not exceed one page.

Note: Remember, you will not be creating the actual code for the method, and you do not have to write pseudocode for both methods.

3. Based on the pseudocode you wrote, **create a flowchart using a tool of your choice for the method you selected**. Your flowchart will help your team communicate how you are planning to develop the software for your client. Your flowchart must be confined to one page. In your flowchart, be sure to do the following:
    - **Include start and end points.**
    - **Include appropriate decision branching.**
    - **Align the flowchart to the check in or check out process.**

    Note: You may draw your flowchart by hand and take a clear picture of it, or you may use a flowcharting tool. Refer to the Supporting Materials section to help you insert your flowchart into your summary report.
4. Based on your software design and development experience, your supervisor has asked you to articulate your programming approach. This will help ensure clarity, consistency, and efficiency among all developers working on this app. Specifically, you have been asked to 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.

What to Submit

To complete this project, you must submit the following:

Pet.java Class File
As part of your tasks for the software development team, create the Pet class in your integrated development environment (IDE). To submit this deliverable, save the Pet class file as Pet.java. Use the Downloading Files From Eclipse Tutorial if you need help with this task.

Global Rain Summary Report
As a junior software developer, you will use the Global Rain Summary Report Template to complete the report. Your report must contain your pseudocode, flowchart, and explanation of the object-oriented programming principles you applied.

Supporting Materials

The following resource(s) may help support your work on the project:

Reading: Pet BAG Specification Document
Review this specification document provided by your supervisor to understand Pet BAG's software requirements. You will use the UML Class diagram portion to implement the Pet class. You will use the requirements in the Functionality section to create your flowchart and write your pseudocode.

Reading: Downloading Files from Eclipse Tutorial
You will write, test, and run your Pet class file using an integrated development environment (IDE). Review this tutorial to learn how to save and export files from your IDE. Important: Do NOT change the names of any class files.

Reading: Creating a Flowchart
These optional tutorials will show you how to make a flowchart using a tool such as Microsoft Word or Lucidchart. You may also choose to draw your flowchart by hand and take a clear picture of it.

- Microsoft Word: Add Shapes
- Lucidchart: How to Make a Flowchart

Reading: Insert Pictures
This optional reading will show you how to insert a picture into a Word document. Use this tutorial to help you add your flowchart into your summary report.

Project One Rubric

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Pet.java Class	Exceeds proficiency by creating a detailed constructor that initializes values for the specified attributes (100%)	Creates a class that meets UML class diagram specifications, including all attributes with appropriate data structures, a constructor, and accessors and mutators (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include appropriate data structures, naming conventions, method structure, return values, or functionality (55%)	Does not attempt criterion (0%)	30
Global Rain Summary Report: Pseudocode	N/A	Writes pseudocode that outlines a plan for developing a method and aligns to software requirement specifications (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include pseudocode clarity, formatting, logic, or alignment to software requirement specifications (55%)	Does not attempt criterion (0%)	15
Global Rain Summary Report: Flowchart	N/A	Creates a flowchart that aligns to software requirement specifications including appropriate decision branching and start and end points (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include logic, representation, appropriate decision branching, or start- and end-point accuracy (55%)	Does not attempt criterion (0%)	15
Global Rain Summary Report: Explanation of OOP Principles	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Explains the application of object-oriented programming principles in a software development process (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include accuracy, depth, or missing principles (55%)	Does not attempt criterion (0%)	30
Articulation of	Exceeds proficiency	Clearly conveys	Shows progress	Submission has	10

Response	in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	meaning with correct grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose (85%)	toward proficiency, but with errors in grammar, sentence structure, and spelling, negatively impacting readability (55%)	critical errors in grammar, sentence structure, and spelling, preventing understanding of ideas (0%)	
				Total:	100%