

# The Perfect Pet (A8)

3/31/2023

Late **79/100 Points**

Attempt 1

Review Feedback  
4/3/2023Attempt 1 Score:  
**79/100**

View Feedback

Anonymous Grading: no

## Unlimited Attempts Allowed

### Details

This project is designed to help you practice a number of aspects of object-oriented programming, specifically the construction of an abstract data type (or two), inheritance from an abstract class or interface, and data persistence through object serialization.

## Overview

You are creating a 'virtual pet' program. The pet object will have a number of attributes, representing the state of the pet. You will need to create some entity to represent attributes in general, and you will also need to create some specific attributes. You will then create a generic pet class (or interface) which has these specific attributes. Finally you will make at least one subclass of the pet class which will be a specific type of pet, and at least one instance of that class.

## Attributes

An attribute is a characteristic of a pet. Each attribute is essentially a list of values. For example, a hunger attribute may have values from "famished" through "content" to "bloated." There should be some way to check the value of the attribute, as well as to increase and decrease the value. Note that you will be expected to create some sort of abstract data type (ADT) to represent the attribute. You are not yet building SPECIFIC attributes (like hunger or happiness) but a type that represents the common characteristics of all attributes. You might use an interface or an abstract class to generate your abstraction, but plan it first as an abstract data type.

## Specific Attributes

Once you have created a generic attribute class, make some subclasses to represent the specific attributes you want your pets to have. If you designed the ADT well, creating specific subclasses should be quite easy. The main method of the specific classes should test the main functionality of the class.

## Making your abstract pet

Now create a pet class that uses the attributes you have generated. This class is also abstract, as it represents just a type of pet. It will include attributes as data members, and it may also have other characteristics. You may also add methods that indicate actions the user can take with the pet, including things like feed and play (which may affect attributes) and perhaps other activities like rename (which might change another data member of the pet) and sleep (which might indicate the passage of time.)

## Build a specific pet class

Finally you should be able to build a specific type of pet (like a lizard or a unicorn) that inherits from the pet class. This should begin (of course) with characteristics derived from the abstract pet, but you could then add new attributes or behaviors specific to your type of pet.

The main method of this pet class should instantiate an instance of the pet and indicate all the things it can do.

## Create a user interface for interacting with the pet.


Build some type of tool for interacting with the pet. At minimum this should allow the user to create a pet, interact with it in the ways you have defined, save its current status for future play (using object serialization) and load a previously saved pet.

## Other artifacts

Please incorporate the following features in your project:

- Documentation in md format
- A UML diagram describing all classes and interfaces
- Please mark abstract classes and interfaces as such on your UML
- A make file including clean and run commands
- An executable jar file.
- Please ensure all documentation is in the jar file
- Also include all java source code
- You will need to create your own manifest as described in class
- Your jar file needs to indicate the appropriate starting class file

## Turning in the project

- This assignment will be turned in through iu github. Please name your repo CSC124000\_spring23\_A8
- If your repo has another name, it will not be graded.
- After you've submitted your assignment in Github, come back to this assignment page and submit the full Github URL (<https://github.iu.edu/username/reponame>  (<https://github.iu.edu/username/reponame>)) for your repo here.

- Your repo must be private. We will not grade projects in public repos, and we will require you to take down any homework assignments in a public repo.
- Please ensure to add all the following are listed as collaborators:
- **Git Collaborators** (<https://iu.instructure.com/courses/2131288/pages/assignments-submission-and-grading>).

Note that your code will be tested with an online plagiarism tool. Please DO NOT turn in work that is not yours. We will know, and we will be displeased.

## BlackBelt extension:

Do it in C++. See the differences between the two languages and how similar some things are.

---

[https://github.iu.edu/parmsing/CSCI24000\\_spring23\\_A8](https://github.iu.edu/parmsing/CSCI24000_spring23_A8)

*You are unable to submit to this assignment as your enrollment in this course has been concluded.*