

CS306 Spring 2009  
Homework Assignment #5 (v2)  
Due: Monday, April 27<sup>th</sup>, 2009 at 4pm  
150 points

**Question 1 (50 points).** (This question requires a more in-depth answer than previous homework questions. I am looking for at least 1-2 paragraphs on each term in the list.) Explain how Perl provides the ability to implement each of the following core OO principles/concepts: Inheritance. Encapsulation. Polymorphism.

### **Program 1 (100 points). Zoo**

*Procedural Note: Please turn in a single file called Zoo.pm for this assignment, rather than using our standard naming conventions. Thank you.*

Your task is to develop an object-oriented model for a zoo, and then write methods which will allow you to populate your zoo with the Animals that we developed in the OO Lab. You should create a Zoo class (which implies a filename of Zoo.pm). Your Zoo class has the following properties.

#### Attributes

- name – a string indicating the name of your zoo. e.g. “Springfield Zoo”. Attribute value must be provided to the constructor. Immutable.
- capacity – the total number of animals that you can host in your zoo. Optional attribute. Default: 10.
- animals – an hash of the animals in your zoo, where the keys are unique IDs assigned by the zoo, and the values are Animal objects. The IDs are integers. Default: empty hash

#### Methods

**Important:** You must use these -exact- names for your methods and write your methods to take the parameters exactly as listed here. This is the API to your Zoo class. I am going to write a driver program (which I will not share with you) which will attempt to use your Zoo.pm class, and add, remove, inventory and listen to animals in your Zoo. I will **even create new types of Animals and make sure that your Zoo can handle them** as well.

- new(\$name) - Creates a new instance of a Zoo with the name \$name. \$name is a string.
- get\_name() - Returns the name of the Zoo.
- get\_capacity() - Returns the capacity of the Zoo.
- set\_capacity(\$new\_capacity) – Sets the capacity of the Zoo to \$new\_capacity if the \$new\_capacity can hold all of the animals currently in the zoo, otherwise prints an error. \$new\_capacity is an integer.
- add\_animal(\$animal, \$animal\_id) – This method adds an Animal to the animals hash with the ID number \$animal\_id if there is available space in the zoo. If there is not space, an error message is printed. \$animal is an object of class Animal and \$animal\_id is an integer.
- remove\_animal(\$animal\_id) – Removes the animal with ID \$animal\_id from the Zoo. If there is no animal by that ID an error message is printed. \$animal\_id is an integer.
- inventory() - Lists the attributes of each animal. For each animal, list its species, ID, name and color. Only list the attributes that are common to all Animals. So a sample list might be displayed like this:

```

Animal ID # 6226
  Species: Horse
  Name: Derby
  Color: Black
Animal ID # 2705
  Species: Cow
  Name: Sunshine
  Color: Black and White
Animal ID # 13
  Species: Snake
  Name: Jake
  Color: Copper

```

- listen() - This method invokes the speak() method for every Animal in the Zoo.
- species() - This method lists all of the species which are represented in the Zoo.

You will then need to write a driver program to test your new class. Your driver program is completely up to you, but at a minimum it should create a new instance of a Zoo, create some Animals, add the Animals to the Zoo, listen to and inventory the Animals in your Zoo, and remove some Animals from the Zoo. So it will start like:

```

#!/usr/bin/perl
use strict;

use Zoo;
use Horse;
use Cow;
use Sheep;

# create a zoo
my $zoo = .....

# create some animals here
# add some animals to the zoo
# maybe try to add too many to make sure error checking is working
# list the zoo inventory
# try to change the capacity of the zoo to some number smaller than number of
animals currently in zoo
# listen to all of the zoo animals
# remove some animals
# list the inventory, make sure animals were removed
# and so on...

```

Turn in just the Zoo.pm class file, not your personal driver program.

### **BONUS POINTS. (20 points)**

Implement the following enhancements:

- Enhance the inventory() method of your Zoo to include attributes specific to each Animal (e.g. a Horse's registry\_id or a Snake's venomous attribute) by adding a print() method to Animal and assuming that any subclass of Animal will also define a print() method (i.e. overload the print() method). (You will need to add the print() method for Horse, Cow and Sheep if necessary, for

testing.)

- Add a second constructor to Zoo which is called like `new($name, $capacity, @animals)`, where `@animals` is a list of `Animal` objects to be added to your zoo.