COIS 2020H-2023 winter

Assignment 1

Due date (Feb, 3rd, at 11:59 pm)

8 marks

**Overview:**

You are going to create a series of objects, that will have a position (which is itself an object). You’re going to store collections of these objects in an array and a list, and you will need to perform basic operations on those structures.

## Required Classes:

1. Class **Position**, which will have

* doubles x, y, and z with setters and getters. These values will be limited to the range +/- 10.0 (i.e., from -10.0 to 10.0). If inputted a value outside that range, clamp to the closest value.
* Method move() which takes doubles dx, dy, and dz and changes the position by + dx, + dy, +dz (it camps to max values of +/- 100 for the edges). The three values dx, dy, and dz can be positive or negative  
  .

1. Class **Animal**, which will have

* Properties: int ID, string name, double age, Position pos.
* Methods. Setters and getters for the fields/properties, and a Move method. Move passes inputted (dx, dy, dz) to the position class (which also will have a move method, the move method in Position should handle the actual implementation of moving things).
* ToString, (public override string ToString()). The Animal class **and** each subclass will implement its own ToString(), and it should just return off all of the properties in some sane format.

1. Animal subclass: **Cat**

* Enum property Breed {Abyssinian, British Shorthair, Bengal, Himalayan, Ocicat, Serval} (if this confused you, look up Enumeration Types on the MSDN[[1]](#footnote-1) C# reference).

1. Animal Subclass: **Snake**

* Properties: double length, bool venomous.

## Main method:

Note: I deliberately throw a lot of randomness on my assignments to simulate the fact that data can be whatever is. For your development/testing work without randomness or use a fixed random seed so it’s always the same, and then test the proper randomness once the regular case works.

Generate a series of animals, 6 each of cats and snakes with random properties, use names from the files provided and avoid duplicates, and load them into an array **and** a list (it can be the same objects in both). [Hint: to use a built-in List you will need to add the following: using System.Collections.Generic; Also, you can check the example in Slide 35 of Week 2 material]

Traverse each collection (the array and the list) and print off the properties of each object. A foreach loop is fine. This should be a loop with a single line.  
  
Move all of the objects with a **random** dx, dy, dz (but only slightly, these can be +/- 5.0 but restrict the final position to +/- 100). [Hint: the following code generates and prints five random numbers between 0 and 10].

var rand = new Random();

for(int i = 0; i<5; i++)

Console.WriteLine(rand.Next(0, 10));

Print off all objects and positions again and highlight in your screenshot that the move works.

…..Check next page

# Submission boxes.

**YOUR NAME:**

|  |
| --- |
| Paste your code for position here, and testing of it correctly setting and clamping values |
| Paste your code for the Animal Superclass here |
| Paste your code for the Cat class here |
| Paste your code for the Snake class here |
| Paste your main code here |
| Show testing of your array here |
| Show testing of your List here |

# Submission guidelines:

Please create a single Word/PDF document where you have a copy of your testing and code pasted using the above table (screenshots are fine, word docs are preferred). Name it for your ***trentusername.docx***. Upload that, along with a zip file (named for your trentusername.zip) containing your visual studio project directories.   
  
Make sure your code formats like code, please use a light theme when copy-pasting. If you face any formatting issues, try to open your .cs files to a docx, or open it using **VISUAL STUDIO CODE** first, then copy the code from there.

Your source code MUST have comments explaining what each class and each method is for, and your testing should cover the major points listed below. You should include some program output (in this case a console window) in your testing document as well, but it should cover major use cases.

1. [Enumeration types - C# reference | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/enum) [↑](#footnote-ref-1)