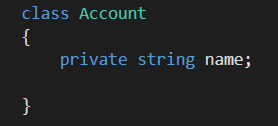
**Lab 3 – Connor Singerline 2/16/2018**  
Intro to OOP

Please answer the following questions in Word (or another text editing program). Upload your completed answers directly to Blackboard this time. Please upload either a Word document or PDF. No other formats will be readable in Blackboard when grading.

1. **Give an example of a getter and setter method for the following class:**



Class Account

{

private string name;

//method that will get the name in the object

public void SetName(string accountName)

{

name = accountName; //store the account name

{

//method that retrieves the account name from the object

public string GetName()

{

return name; //returns name’s value to this method’s caller

}

}

1. **How would you instantiate an object or type Account?**

Account myAccount = new Account();

1. **Reformat the following names using camel-case formatting:**

Answer: variable names begin with an initial lowercase letter ~ camel case

Where as class, property, and method names begin with an initial uppercase letter ~Pascal Case

Al Dente

= alDente

Alphabet Soup

=alphabetSoup

Cuyahoga Community College

= cuyahogaCommunityCollege

number 10

=number10

1. **Parameters are also considered what? Where can parameters be used?**

Parameters are also considered local variables. Parameters must specify a type such as string, int, etc, followed by a parameter name. Parameters can be used in a methods body and are declared inside a class definition.

1. **What are two values we can use for access modifiers? What does this indicate?**

private and public; This is known as information hiding – the instance variable set to private is hidden and can be used only in the class assigned. Methods (and other class members) that are declared public are “available to the public”. These can be used by methods (and other members) of the class in which they’re declared. They can also be used by the class’s clients – that is, methods (and other members) of any other classes.

1. UML diagrams help us to think through what a class will have (properties/attributes and methods/operations) in it before we code it out. Consider the following classes. **Add two or more attributes and operations to both classes.** Think of a real-world application that will use these classes. You do not need to create an image...just fill them in here.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Person** | | - name : string - zipCode : int  - dateOfBirth : string  - hairColor: string | | + Talk (string sentence, int volume)  + Move (double direction) : double  + Dance (double direction) : double  + Change Hair Color (hairColor string, durationMinutes int) | | |  | | --- | | **Cup** | | - volume : int  - type : string //glass, plastic, copper  - color : string | | + Fill (int volume) : int  + Empty (int volume) : int  + Stack (int numberOfCups) : int | |