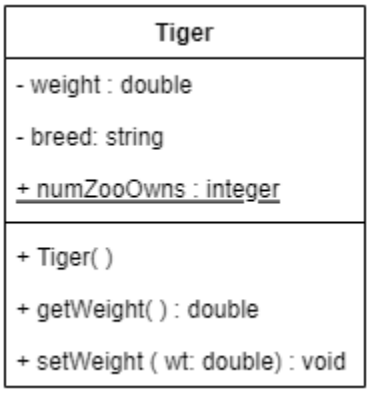
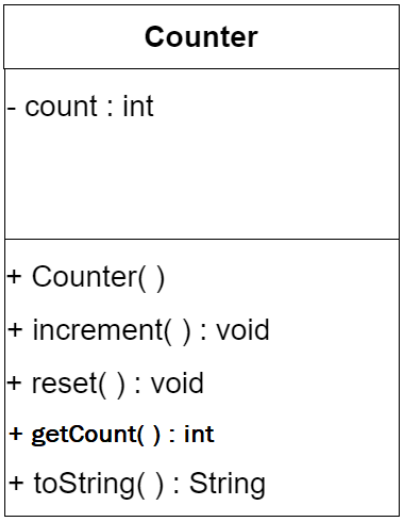
Original Partner **SOLUTION** Student #	Obstacle 5 COMP 10062	Paired Partner ("grader") Student # **Odds & Ends**	SCORE OBTAINED 5

Answer each question in the space provided. There's a ZIP file available in order to help you get these questions right!

1	What is the Java line of code which will create a public *constant* of: 2.7182818	**public static final double E_CONSTANT = 2.7182818;**

2	Draw the UML diagram which represents the code below:
	
	public class Tiger {
	private double weight;
	private String breed;
	public static int numZooOwns;
	
	public Tiger() ;
	public double getWeight();
	public void setWeight(double wt);
	}
		

3.	Declare an array called, **goals**, and populate it with 50 ***Counter*** objects all set to 1. Do NOT add any new instance vars or methods to the Counter class shown below. **Note: When a new Counter is constructed its count is set to zero** 	**Counter[ ] goals = new Counter[50];** **for (int c=0; c< goals.length; c++) {** **goals[c] = new Counter( );** **goals[c].increment( );** **}**

4.	Suppose the Counter array elements all have varying counts. Output each of the 50 counter's current count value using an enhanced for loop.	**for (Counter c : goals) {**
		System.out.println(c.getCount());
		}
		
		// ========== an aside …. ==========
		public static void main(String[] args)
		{
		Die [] dice = new Die [1000000];
		for (Die d : dice) {
		d = new Die();
		}
		
		The code above does not create Die objects in each array element because of how the enhanced for loop works in Java. In the enhanced for loop, the variable d is not a reference to the actual array element, but rather a temporary variable that holds the value of each element in the array during each iteration.
		
		Assigning a new value to d does not modify the corresponding array element. The code is assigning a new Die object to the temporary variable **d**, but it does not update the array element dice[i]. As a result, all elements of the dice array remain null.

5.	a) Write the java code which would output ONLY the current count of the 49th element. b) Write a standard for loop which will reset all of the 50 counts to 0. (i.e. the reset state)	**System.out.println (goals[48].getCount());** _____ the 49th element's current count **for (int n = 0; n < goals.length; n++) {** **goals[n].reset( );** **}**
6.	Declare IN ONE LINE ONLY a 4-element boolean array called, b to all true values.	**boolean[ ] b = { true, true, true, true };**